TK 90X: MICROCOMPUTADOR...

Para você que quer explorar todo o potencial do seu micro TK 90X como ferramenta de trabalho e como fonte de lazer criativo, a Cibertron tem novidades. "O Artista", "Controle de Estoque", "Banco de Dados"* e "Planilha 90X"*, ao preço de Cz$ 99,90 cada, são os novos programas que a Cibertron está lançando. Aproveite e conheça também os já tradicionais jogos e aplicativos Cibertron para o TK 2000 e TK 85.

...E NÃO APENAS VIDEOGAME.

S O F T W A R E

METEOR STORM — TK 90X · SIMULADOR DE VÔO — TK 2000 · XADREZ — TK 2000 · DRÁCULA — TK 90X · SIMULADOR DE VÔO — TK 90X · BOLO — TK 2000 · CAMPO MINADO — TK 90X · FALCONS — TK 2000

UFO — TK 85 · SABER — TK 85 · PINBALL — TK 85 · TK-WORD — TK 85 · ROBOT — TK 85 · ELIMINATOR — TK 2000 · EGGS — TK 2000 · CEILING ZERO — TK 2000

É C I B E R T R O N

Adquira o Software Cibertron nos revendedores abaixo. Caso você não encontre o programa desejado, peça-o à Cibertron Eletrônica Ltda. Caixa Postal 17005 – CEP 02399 – SP, anexando ao pedido um cheque nominal (remessas em três dias úteis). Solicite, gratuitamente, a nossa lista de programas.

Revendedores autorizados:

amarosom
R Pamplona 982 Tel 288 1900
Al Jaú 661 (esq Pamplona)
Tel 284 9644

Tel. (011)255 7653

SANTO ANDRÉ
SHOP audio & video
R Gal Glicério 430 Tel 444 6055

* Disponíveis a partir de 15/05/86.

ÍNDICE

**O tema da capa deste mês trata sobre as linguagens de programação tentando esclarecer as dúvidas dos profissionais e hobbistas sobre as utilizações e características das mais conhecidas.
Capa: Hugo Faleiros e Eduardo Previdelli.**

EXPEDIENTE

DIRETOR RESPONSÁVEL
Abraham Poppovich

EDITORA
Ana Lúcia de Alcântara Oshiro (M.T. 14495)

REDAÇÃO
Mônica Rocha
Tânia M. Cristina Batista (Secretária)

CPD
Marcos Lorenzi
Paulo Panagiotis Alvanos
Elaine Kelly Lima

ASSESSORIA TÉCNICA
Wilson José Tucci

CORRESPONDENTES
Fátima França — Rio de Janeiro

COLABORADORES
Aroldo Possuelo de Carvalho, Cesar de Afonseca e Silva Neto, Christiano Coutinho Nassur, Daniel José Burd, Dirceu Teixeira, Fernando Carvalho Filho, Paulo Sergio Naddeo D. Lopes, Ricardo Urbini, Zilah Mimiazaki.

MARKETING
Aurio José Mosolino (supervisor)
Maria de Fátima Sales

ASSINATURAS
Marli Mantovani

CIRCULAÇÃO
José Aparecido Bueno

ADMINISTRAÇÃO
Cleusa Ap. S. Malian

DIAGRAMAÇÃO, ARTE, FOTOCOMPOSIÇÃO, FOTOLITO
Coruja Fotolito Ltda.

IMPRESSÃO
Bandeirantes S/A - Gráfica e Editora

MICROHOBBY é editada mensalmente por Microdigital Eletrônica Ltda. (DIVISÃO MICROHOBBY).
Endereço para correspondência:
Rua do Bosque, 1.234
Caixa Postal 54096 — CEP 01136
São Paulo — SP — Fone: (011) 825-4066

Para solicitar assinatura anual utilize o encarte nesta Revista e pague em qualquer agência do Banco Bradesco ou envie cheque nominal em nome da Microdigital Eletrônica/ Divisão Microhobby.

MICROHOBBY 34

AGOSTO/86

EDITORIAL

Esta edição de MICROHOBBY estará presente no maior evento da área de informática no Brasil. O Info/86 que leva o lema da ***informática como instrumento de desenvolvimento*** apresentará com certeza (no Riocentro, Rio de Janeiro), a tendência dos lançamentos voltados aos equipamentos de 16 bits, com a maioria dos fabricantes nacionais apresentando a sua força e cada um, a seu modo, tentando mostrar ao público a consolidação de sua Empresa no mercado.

O XIX Congresso Nacional de Informática não trará muitas inovações em sua estrutura organizacional, pois estará recheado de palestras técnicas e programações especiais, com convidados internacionais e debates que abordam assuntos referentes à política nacional para o setor. Quanto à Feira, aguarda-se uma maior área dedicada à exposição e melhorias na infraestrutura do Riocentro - reivindicação antiga dos expositores e que vem sendo prometida pelos organizadores para este ano.

Acreditamos que um dos temas a serem debatidos no Congresso será, provavelmente, aquele relacionado à política de software. Este assunto vem tendo bastante espaço na imprensa diária e tem levado a inúmeras discussões, tanto a nível governamental como por parte de usuários e empresários do setor. Vamos aguardar e ver como serão os debates a este respeito e torçamos para que algo de novo seja gerado nestes debates.

A MICROHOBBY trará, na próxima edição, os resultados das palestras do XIX C.N.I. e publicará a tendência das discussões notadas nas mesmas. Traremos também, os principais lançamentos da Feira.

Para esta edição escolhemos, como tema principal, o artigo "Linguagens de Programação", cujo assunto continuaremos explorando em edições futuras. Mostramos também um lançamento da Microdigital, que será de bastante utilidade para os usuários do TK-90X. O ***Light-pen*** é um periférico há muito esperado por estes usuários e que agora se encontra no mercado.

Finalizando, gostaria de deixar de lado as formalidades e colocar aqui um recado muito especial a todos os nossos leitores:

— "Obrigado pela receptividade e apoio que vocês têm nos dado. As críticas e novas idéias que nos enviam são a nossa "principal inspiração" (não é demagogia, não!).

A vocês, associados do nosso ***Clube de Usuários,*** deixamos um especial agradecimento pelas respostas enviadas à nossa pesquisa.

Até mais!

* Envie-nos, de volta, a pesquisa que passamos a publicar nesta edição, sobre a opinião de vocês a repeito das seções da MICROHOBBY.

Ana Lúcia de Alcântara Oshiro

CARTAS

Microhobby
Rua do Bosque, 1234
São Paulo, 01136

Dúvidas técnicas sobre programas e equipamentos apenas por carta. Não respondemos por telefone.

Prezados Senhores,

Gostaria de saber como é que eu tenho de proceder para obter o livro Entendendo a ROM do TK-2000II; *livro comentado na Revista de número 30, na seção de cartas. Gostaria também, que me ajudassem a encontrar os seguintes livros:*
- Synertek 6500 Hardware Manual
- 6502 Assembly Language Programming/Lance A. Leventhal
- 6502 Assembly Language Subroutines/ Lance A. Leventhal

Marcelo Caetano de Oliveira
Águas de São Pedro, SP

Caro Marcelo,

Para aquisição do livro ***Entendendo a ROM do TK-2000,*** basta escrever para Caixa Postal 54096 enviando um cheque nominal à Microdigital Eletrônica/Divisão Microhobby, no valor de Cz$ 75,00.

Os livros que gostaria de adquirir, você poderá procurá-los na Livraria LITEC em São Paulo. Basta procurar o anúncio na revista.

Caros Senhores,

Sou assinante da Revista MICROHOBBY à qual considero de excelentes qualidades técnicas e editoriais.

Gostei muito do curso de Assembler para o micro TK-90X iniciado em abril.

Outra coisa que me interessa muito é um artigo sobre a light-pen para o TK-90X pois pretendo adquirir uma assim que chegar às lojas.

Sotero Luiz da Silva
Santos, SP

Caro Sotero,

Agradecemos os elogios destinados à nossa equipe, ficamos muito gratos em saber que nosso trabalho está atendendo às necessidades de nossos leitores.

Na edição 34, sairá um artigo falando sobre a light-pen, que já está disponível no mercado, explicando seu uso e quais as suas utilidades. Aguarde!

Caros Senhores,

Gostaria de saber se o programa Mansão de Sherlock Holmes *da MICROHOBBY 26, página 45, roda no TK-2000 II.*

Eu digitei o programa todo, quando fui rodar nada aconteceu e ao listá-lo, vi que só ía até a linha 1300. Porque isto aconteceu?

Edison Pires
São Paulo, SP

Caro Edison,

O programa **Mansão de Sherlock Holmes** desenvolvido para o Apple pode rodar no TK-2000 II, claro que há necessidade de se fazer algumas modificações.

Quanto ao fato de você ter digitado o programa e este não rodou e, na hora da listagem, só apareceu até a linha 1300 porque o programa é muito extenso. A primeira página de vídeo ficou lotada e o restante invadiu a segunda página. Como você não uniu a segunda com a primeira, através do comando MP, a outra parte do programa foi perdida.

No Apple também existe estas duas páginas de vídeo mas elas já estão unidas formando uma só. Observe no Manual Técnico do TK-2000, página 115 e veja o Mapa da Memória.

Para conseguir a gravação de informações alfanuméricas é necessário usar determinados macetes que foram editados na MICROHOBBY 23 com o título "ROTINA DE GRAVAÇÃO DE DADOS ALFANUMÉRICOS".

Além disso, temos publicado, periodicamente na MICROHOBBY, resenhas sobre os softwares para os diversos equipamentos.

Quanto a livros que detalhem este assunto devem ser procurados na livraria LITEC, onde podem ser encontradas literaturas técnicas de todos os tipos. O endereço é Rua dos Timbiras 257, CEP 01208 São Paulo SP / Caixa Postal 30869, telefone: (011) 222-0477. Vendas pelo reembolso postal VARIG, solicite catálogo de seu interesse.

Prezados Senhores,

Possuo um TK-2000 Color 128 K e sou assinante da revista. Toda matéria que tem saído e que é compatível com o meu micro dou uma atenção especial.

No entanto, até agora não fui feliz com os programas que tratam de criação e manipulação de arquivos com fita. As literaturas e programas que consigo na praça quase todas são para discos.

Portanto, caso os senhores tenham bons livros ou bons programas para me ajudar a entender "arquivo em fita", eu ficaria muito grato com as indicações.

Tulio Metzkeir
Belo Horizonte, MG

Caro Tulio,

A utilização de fita no TK-2000 para armazenamento de dados oferece algumas desvantagens, tais como: organização e acesso seqüencial e armazenamento somente de dados numéricos.

CARTAS

Os dois comandos oferecidos pelo equipamento são:
- STORE - para a gravação;
- RECALL - para leitura.

O armazenamento é feito através de matrizes, que devem ser dimensionadas igualmente na gravação e na leitura.
(ativa PRINTER) e CALL 1305 (desativa PRINTER) usamos a impressora.

No TK-2000 usa-se o INVERSE e NORMAL para substituir FLASH e, o IN, utiliza-se um CALL correspondente para substituí-lo.

Se o FLASH for retirado, o programa funcionará normalmente, apenas a informação desejada não piscará na tela, já o IN poderá causar uma parada no programa.

Prezados Senhores,

Solicito que me respondam estas pequenas dúdivas:

1. *o comando PR que combinado com os números 0 e 1 acionam e desligam a impressora, é compatível com o computador TK-2000? Se não, qual outro comando o substiui?*
2. *que outros comandos do TK-2000 substituem os comandos FLASH e IN , no computador Apple II Plus?*
3. *para que servem os comandos que eu especifiquei anteriormente. Diga-me o que eles fazem. Se eu retirar estes comandos de um programa, o mesmo rodará normalmente?*

Grinaldo Lopes de Oliveira
Salvador, BA

Caro Grinaldo,

Devido ao Apple apresentar slots que permitem a conexão de vários periféricos, entre eles a placa PRINTER, pode-se acionar diretamente a impressora usando os comandos PR 1 e PR 0.

Como o TK-2000 não apresenta slots, então é necessário primeiramente carregar o software de impressora e então, através dos comandos CALL 1302 (ativa PRINTER) e CALL 1305 (desativa PRINTER) usarmos a impressora.

No TK-2000 usa-se o INVERSE e NORMAL para substituir FLASH e, o IN, utiliza-se um CALL correspondente para substituí-lo.

Se o FLASH for retirado, o programa funcionará normalmente, apenas a informação desejada não piscará na tela, já o IN poderá causar uma parada no programa.

Prezados Senhores,

Faz tempo que desejava enviar algumas idéias que poderiam ser de interesse para a revista e seus leitores. Desejaria também colocar alguns tópicos relacionados com o TK-2000.

Uma observação: a revista MICROHOBBY em mais alguns números, irá fazer três anos. De fato, o material que ela tem publicado é farto, variado, e útil. Poderia ser aproveitado o número 36 para editar um índice temático, no estilo das revistas científicas, classificando os artigos dos números 1 a 36.

Gostaria de saber se existe um Editor de Texto (não simplesmente de linhas) para o TK-2000, e quem o fornece. Se a resposta for negativa, não estaria na hora de alguma empresa assumir o desafio? Pode ser dito que o TK-2000 carece de minúsculas, mas isto ë aparente (só) no caso de querer imprimir um texto via impressora; o código ASCII para minúsculas é exatamente aquele das maiúsculas, somando 32 (por exemplo, ASC(A) = 65, e ASC(a) = 97...).

Existem jogos e aplicativos em fita para o TK-2000. Não existem também em disco? Se a resposta for negativa, não seria interessante que as produtoras de software editem tanto em fitas como em disco?

No caso do TK-3000, ele é compatível com aplicativos como o Magic Window, *admitidos pelos Apple?*

E por fim, desejaria pedir a maior: Negócios são negócios, tudo bem; mas o TK-2000 é uma opção linda demais frente aos Apple, como podemos abandoná-la...

Um abraço (simbólico), e meus parabéns pela sua excelente revista.

Juan Carlos Ceballos
Campina Grande, PB

Caro Juan,

Ficamos profundamente agradecidos, pelo fato de lembrar que a MICROHOBBY em breve irá completar três anos de existência.

Até o momento a Microdigital não desenvolveu nenhum Editor de Texto para o TK-2000 apenas o editor de linhas, talvez exista no mercado, mas não temos certeza.

A possibilidade de se obter minúsculas no TK-2000, é através de um programa publicado na edição 26, página 14.

Existe no mercado uma grande variedade de jogos e aplicativos para o TK-2000 tanto desenvolvidos pela Microdigital como por outras empresas de software.

O Magic Window é compatível com o TK-3000. Neste equipamento, ele funciona perfeitamente.

Respondendo a sua última indagação: já existem vários programas sendo comercializado em disco para o TK-2000.

Prezados Senhores,

Tenho duas dúvidas:

É possível, via software, criar novas cores no TK-2000, já que são tão poucas as cores que ele possui ou apenas via hardware. Pergunto, porque outros computadores que possuem o mesmo microprocessador, o 6502, têm várias cores?

Há possibilidade de colocarmos a rotina do TK DOS 3.3 do TK-2000 acima da MP, ou seja, em cima da ROM, com o intuito de liberar mais memória RAM a ser gravada em disco, pois há vários programas em fita que não consigo passá-los para disco.

Hernando Cipriano da Silva
Rio de Janeiro, RJ

Caro Hernando,

Aguarde a edição 34, onde será publicado um artigo sobre como utilizar os 64K do seu TK-2000.

O TK-2000 foi desenvolvido para trabalhar com seis cores no modo de alta-resolução, incluindo o preto e o branco. Mas, através de um processo DITHERING (excitação) é possível produzir, artificialmente, onde este número pode ser aumentado consideravelmente. Esta informação encontra-se no Manual Técnico, páginas 65 e 66.

Existem no mercado nacional vários micros que possuem o mesmo microprocessador, mas estes diferem quanto à sua configuração interna que varia de um fabricante para o outro.

O TK-2000 foi desenvolvido para trabalhar com sete cores tanto em baixa como em alta-resolução. Não havendo a possibilidade de misturar as cores.

O Apple possui o dobro de cores que o TK-2000, porque é possível fazer combinações entre elas o que aumenta em dobro sua capacidade de cores em baixa e alta-resolução.

Experiências e Recados

O nosso clube é muito novo, tivemos a idéia justamente lendo a MICROHOBBY. Infelizmente ainda não temos muitos sócios pois nosso nome foi publicado recentemente.

Até agora, por intermédio desta revista, recebemos inúmeras cartas.

A nossa idéia foi a seguinte: retiramos o nome de alguns clubes de usuários dessa revista, e mandamos cartas convidando-os para participar do nosso clube. Muitos deles aceitaram. Por isso, se possível, gostaria que me enviassem, por obséquio, mais alguns nomes de clubes de usuários.

Estamos trocando programas. Como vocês já devem saber, nós enviamos um jornalzinho para os sócios. Nele, enviamos uma série de dicas, programas etc. Inclusive, todos os sócios nos mandam programas, enfim, coisas do nosso interesse.

A parte mais importante da MICROHOBBY, na minha opinião, é justamente, **O Clube de Usuário.** É nele que todos se unem e comentam o conteúdo da Revista.

Se vocês quiserem poderemos sempre que puder, lhe enviar um relatório sobre as atividades realizadas no SOFTCLUB.

Softclub
R. Pascoal Ranieri Mazzilli, 634 - V. Industrial
03257 - São Paulo, SP

Caros Sidnei e Marcelo,

Obrigado pelo seu recado aos nossos sócios. Gostaríamos que dessem uma atenção especial na reportagem da edição 33 sobre as finalidades de nosso clube de usuários.

Estamos muito interessados em receber um relatório contendo informações sobre as suas atividades, pois com isso poderemos publicá-las em nossa revista, como informação aos nossos sócios.

Desejo receber informações sobre livros e grupos de usuários do Espírito Santo. Se possível, em Cachoeiro de Itapemirim ou nas cidades vizinhas.

Alcir Cesar da Silva
R. Manoel D. Monteiro, 13 - Nova Brasília
29300 - Cachoeiro de Itapemirim, ES

Caro Alcir,

O seu recado está dado. Espero que receba muitas cartas.

Ana Lúcia, estou enviando-lhe esta carta, para que você me dê alguma sugestão para trocar programas por correspondência, pois é difícil mandar fitas, além de arriscado.

André Dastre Vilas Boas
R. Machado de Assis, 16/33
17043 - Bauru, SP

Caro André,

Quando um usuário deseja trocar programas com outros, nós deixamos que este realize seu objetivo da melhor maneira possível.

Concordo que seja arriscado mandar fitas pelo correio porque corre-se o risco destas extraviarem. Mas o nosso serviço de correio melhorou muito, e tenho certeza que você terá sucesso nas suas correspondências.

Tente se comunicar com algum usuário que já tenha feito este tipo de transação para obter maiores explicações de como mandar programas por correspondência. Aqui está dado o recado aos outros usuários para que orientem o André.

Aproveitando a oportunidade, gostaria que vocês colocassem um aviso para o Klaus, de São Paulo, dizendo que eu quero trocar programas com ele, mas que mande uma outra carta porque eu perdi o endereço dele.

Carlos Henrique Tavares Mero
R. Antonio Procópio, 716 - Farol
57000 - Maceió, SP

NOVOS SÓCIOS

TK 85 E COMPATÍVEIS

Aldo Jorge Nakamura
Av. Paulista, 854 - 12º andar
01310 - São Paulo, SP
(também possui um Apple)
Área de interesse: gráficos e aplicativos.

Cyro Alexandre C. de Leone
Rua Manaus, 38/112 - Água Fria
02335 - São Paulo, SP
(também possui um TK-2000)
Área de interesse: geral.

Yuri Alexei Cairof Sampaio
Rua Paulino da Silva Lavandeira, 204
17500 - Marília, SP
Área de interesse: jogos, educativos, programas.

Gilson Pereira
Rua Duque de Caxias, 188
96200 - Rio Grande - RS
(também possui um TK-90X)
Área de interesse: organiza um clube de usuários em sua cidade.

TK 90X E COMPATÍVEIS

Fernando Keiji Yoshimo
Rua Cantarricos, 83 - Imirim
02540 - São Paulo, SP
Área de interesse: jogos, assembler, dicas de eletrônicas, aplicativos.

Marcelo Dionisio Penalva
Av. 64, nº 2493 - J. Panorama
13500 - Rio Claro, SP
Trabalha com TK, Apple, TRS-80

Daniel Israel Mignone
Rua Gomes Carneiro, S11/104 - Ipanema
22071 - Rio de Janeiro, RJ
(também possui TK-85 e TK-2000)

Edson Yuichi Suzuki
Rua Agostinho Paporalli, 1116 - V. Oliveira
08700 - Mogi das Cruzes - SP
Área de interesse: assembly, educação

TK 2000 E COMPATÍVEIS

Carlos Henrique Tavares Mero
Rua Antonio Procópio, 716 - Farol
5700 - Maceió, AL
Área de interesse: jogos, música, gráficos, aplicativos

Fritz S. Arnold
Rua Dr. Paulo Duarte, 11 - Pirituba
05165 - São Paulo, SP
(também possui um TK-85)
Área de interesse: aplicativos, hardware, jogos, LM

Guilherme Lopes
Travessa Narcejo, 138 - Lins
20710 - Rio de Janeiro, RJ
Área interesse: engenharia, arquitetura, desenho técnico, jogos

APPLE E COMPATÍVEIS

Eduardo Marchiori
Rua Mariano Procópio, 58
01548 - São Paulo, SP

Aldo Jorge Nakamura
Av. Paulista, 854 - 12º and.
01310 - São Paulo, SP
(também possui um TK-85)
Área de interesse: gráficos, aplicativos

Guilherme João Muller
Rua Canadá, 213
80000 - Curitiba, PR
(também possui um TK-2000)
Área interesse: linguagem de máquina

MICROHOBBY PRESS

M.P. 003

Informática: um instrumento de desenvolvimento.

Informática, desenvolvimento, cidade maravilhosa e tecnologia. Estes são os temperos básicos da VI Feira e do XIX Congresso de Informática deste ano. Preocupados em discutir os rumos da informática no Brasil, apresentar a consolidação do setor e a opção da nossa indústria pelo mercado voltado a aplicações profissionais, os seus organizadores abrirão as portas do Riocentro cercados de enorme expectativa em torno do sucesso do maior Evento de informática no País.

Ana Lúcia de Alcântara

Mostrar em que medida a informática está contribuindo para o desenvolvimento do País nos diversos segmentos da economia é um dos principais objetivos dos organizadores da VI Feira Internacional e do XIX Congresso Nacional de Informática que serão realizados, em paralelo, nos dias 18 a 23 de agosto, na cidade do Rio de Janeiro.

O maior evento da área de informática no Brasil, considerado um dos quatro mais importantes no setor em todo o mundo, tem este ano um tema que segue a mesma linha da temática do Informática do ano passado - realizado em São Paulo. ***Informática, um instrumento de desenvolvimento*** dá continuidade ao ***Informática, Progresso e Desenvolvimento*** trazendo os resultados alcançados na área, nos setores científico-tecnológico, econômico e político, após um ano da Nova República.

Enquanto o Informática/85 pretendia discutir, em seu Congresso, os aspectos políticos que envolviam o setor, até então, como Legislação de Software, Registros das profissões ligados ao Processamento de Dados e Avaliação dos resultados obtidos no primeiro ano da Lei de Informática, o Informática/86 quer apresentar ao público a consolidação do setor, os índices de desenvolvimento alcançados por outros, a posição dos usuários na Constituinte e principalmente, "mais aplicações da tecnologia nos diversos setores da economia e muito menos discussões acadêmicas".

Este novo posicionamento se deve, segundo os organizadores do Evento, aos resultados obtidos na pesquisa realizada pela SUCESU - Sociedade dos Usuários de Computadores e Equipamentos Subsidiários do Rio de Janeiro, junto aos participantes do Informática/85 que "permitiram conhecer toda a tendência de interesses, principalmente dos congressistas". Esta tendência leva à preferência das aplicações que se dividem em diversos segmentos: banco de dados, desenvolvimento, implantação e manutenção de sistemas, administração de informática, entre outros.

A preocupação com a opinião dos participantes neste Evento trouxe, além de alterações temáticas no Congresso, algumas inovações também na Feira. Este ano, a VI F.I.I. conta pela primeira vez, com a formação do Conselho de Expositores, integrado por todos os segmentos presentes e que visa, principalmente, eliminar problemas de ordem operacional que, em realizações anteriores, devido à localização e estruturação do Riocentro, trouxeram alguns prejuízos aos expositores e participantes do Congresso.

O Evento contará também com o apoio do Governo do estado que o vem considerando "prioridade máxima dentro da linha adotada pela sua administração em fornecer apoio irrestrito ao setor de informática na região". Para isto, o Governo do Rio de Janeiro destinou recursos que chegam a Cz$ 5 milhões para a realização de obras no Riocentro que visam à melhoria e sinalização das vias de acesso ao local.

A Feira

Ocupando uma área de 25.000 m², a VI Feira Internacional de Informática contará com a presença - de acordo com dados divulgados pela Foco - Feiras, Exposições e Congressos Ltda. (empresa organizadora) - de aproximadamente 300 mil pessoas e com cerca de 300 expositores nacionais e internacionais, que estarão distribuídos nos diversos setores da exposição. Equipamentos e serviços de informática, mídia eletrônica, telemática, automação bancária e de escritório, serviços e equipamentos de computação, periféricos e componentes de eletrônica digital, telecomunicações, microfilmagem, móveis, equipamentos e artigos para escritório e pesquisa e desenvolvimento são as diversas atrações desta Feira.

Porém dois setores em especial receberão benefícios importantes na Exposição de agosto. O primeiro setor é o de pequenas e microempresas, que terão incentivos financeiros para participarem da Feira, através de um projeto especial criado pela Foco e pela Sucesu, subsidiado pelo Banerj - Banco do Estado do Rio de Janeiro e pela Flupeme. Dessa forma, foi criada uma área específica para software, agregada por software básico, de apoio e aplicativo que é tormada por boxes/balções já montados, em lugar dos caríssimos estandes. Espera-se segundo a Foco, a participação de cerca de 50 microempresas. Haverá também, além do Banerj, o Banco de Desenvolvimento do Rio de Janeiro que apresentará o seu Projeto Pró-Info de benefícios (com financiamentos de 150 mil OTNs, juros de 6% ao ano e prazos de até oito anos) aos micros, pequenos e médios empresários.

O outro setor é o de Pesquisa e Desenvolvimento, que teve a sua área ampliada dos 1500 do ano passado para os dois mil deste ano, assim como o aumento do número de instituições e centros de pesquisas participantes (de 25 para 40), que pretendem mostrar não só o que fazem em P&D, mas o que a informática está fazendo pela pesquisa e desenvolvimento em outros campos de alta tecnologia como a agricultura, biotecnologia, energia, sociologia, medicina, etc. Esta área já tem confirmadas a presença de 32 instituições e trará das universidades aproximadamente 300 pessoas, além de pequenas empresas que trabalham com tecnologia emergente envolvendo um orçamento estimado em Cz$ 3.500.000,00.

A Feira trará com certeza, vários lançamentos e a tendência dos lançamentos, segundo muitos prevêem, se centrará no campo dos 16 bits, que buscarão atingir o interesse do público mais especializado.

O Congresso

Fazer um Congresso de Informática muito mais pragmático que teórico é o objetivo da SUCESU/RJ para este ano. Com este intuito, os organizadores deste XIX Congresso apresentarão diversas pales-

tras técnicas voltadas aos profissionais que falarão de trabalhos nas áreas de controle de processos, automação industrial, comercial e de serviços, apoio à decisão, entre outros. Ao todo foram selecionados 90 trabalhos que têm temas como "Formação e Aperfeiçoamento de Recursos Humanos em Informática"; Teleinformática; Inteligência Artificial e Robótica etc.

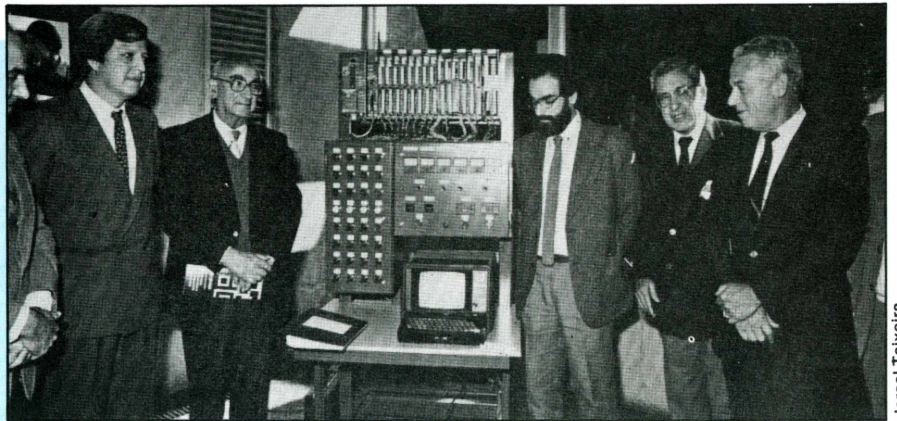
O Telecongresso, que foi uma das principais atrações do Congresso realizado em São Paulo, em 85, será mantido este ano e trará debates que discutirão, entre outros assuntos, a proteção do consumidor de informática (o usuário); informação, informática e a constituição; os modelos de desenvolvimento da informática e a nova ordem econômica e informática.

Um dos destaques neste Congresso será, com certeza, a programação especial dedicada à área de microinformática. Este espaço foi criado, de acordo com os organizadores, devido às características próprias deste setor, pois sentiu-se a necessidade de compatibilizar o ***Microinfo*** às expectativas do seu público alvo (bastante heterogêneo): os usuários de microcomputadores.

Para atender este público heterogêneo elaborou-se uma programação formada de palestras sobre as diversas famílias e linhas de equipamentos existentes e a respeito do software e aplicações do micro, apresentando uma visão conceitual das principais aplicações e a descrição dos pacotes disponíveis para os diversos equipamentos.

Haverá ainda no ***Microinfo*** outros eventos que contarão com a presença de convidados do Exterior. Entre estes, destaca-se a palestra sobre "Microinformática - presente e futuro". ◇

Telex

Almarés, empresa especializada na revenda de produtos de informática na cidade de São Paulo, estará ampliando sua área de atuação. Com nova direção técnica, ocupada por Marcelo Leon de Oliveira, um jovem executivo de 30 anos com currículo extenso e formação especializada na área de eletrônica, a Imarés passará a incrementar a sua recém-fundada Divisão de Desenvolvimento de Projetos Especiais, que exigiu da Empresa um investimento de Cz$ 800 mil distribuídos entre pesquisa, aquisição de equipamentos e contratação de pessoal especializado. ◇

Israel Teixeira

O Controlador Lógico Programável doado ao SENAI, pela Metal Leve

Mais um incentivo ao Projeto do SENAI

O Projeto de Informática do Senai - Serviço Nacional de Informática, cuja maior atuação tem sido notada na regional de São Paulo, recebeu mais um incentivo para sua consolidação, durante o último mês de julho. A entidade recebeu, como doação, um equipamento no valor de Cz$ 600 mil da Metal Leve, que será utilizado na Escola Senai "Antonio Noschese", da cidade de Santos.

O equipamento, um Controlador Lógico Programável (CLP), acompanhado de painel simulador e terminal de programação, será utilizado no Laboratório de Controladores Lógicos da Escola de Santos, um dos quatro laboratórios onde se desenvolvem o treinamento de especialização em circuitos digitais e microcomputadores, destinado a técnicos de nível médio e superior, de empresas usuárias de sistemas digitais para controle de processos, filiadas à Entidade.

O Projeto de Informática do Senai foi iniciado em 84 e o laboratório de circuitos digitais e microcomputadores inaugurado no ano passado. ***A.L.A.*** ◇

Pacotes específicos para PC e Apple

A Compusoft, empresa paulista especializada no desenvolvimento de sistemas e atuando desde 83 neste setor, está colocando no mercado, através da rede de comercialização da Compushop (São Paulo - Capital, Bauru e Ribeirão Preto), um pacote de programas com aplicações específicas, desenvolvidos inicialmente, para alguns de seus clientes.

Administração Escolar, Controle de Distribuidora de Títulos e Valores Imobiliários, Clube de Investimentos, Controle de Convênios Médicos, Fluxo de Caixa e outros sistemas administrativos são alguns dos programas que fazem parte deste pacote.

Todos os programas da Compusoft são desenvolvidos em dois sistemas operacionais básicos (CP/M e MSDOS) e se apresentam, segundo Luis Branco, gerente geral da Empresa, para o usuário de uma maneira "bastante amigável", pois são estruturados em ***menus*** que fornecem a este, toda a informação necessária, com leitura de tela facilitando dessa forma, o entendimento de programa e o trabalho do mesmo.

Os softwares desenvolvidos inicialmente para clientes específicos, têm tido, conforme ressaltou Branco, uma procura bastante acentuada por parte dos usuários, pessoas físicas. "Um destes produtos" - exemplificou, "é o de Valores Mobiliários e o Clube de Investimentos".

A Compusoft tem atuado também, na área de treinamento voltado à microinformática para profissionais e iniciantes em computação, assim como fornecendo cursos fechados às empresas e na área de consultoria, fornecendo aos clientes, serviços especializados como a conversão de arquivos Apple-PC e projetos de sistemas multiusuário. ***A.L.A.*** ◇

MICROHOBBY PRESS

Muitos produtos e poucas novidades

Com a aproximação da VI Feira Internacional de Informática, à III Expo Soft não mostrou muitas novidades em matéria de lançamentos.

Realizada entre os dias 25 e 27 de junho, no Centro de Convenções Rebouças (em São Paulo), a III Exposição Nacional de Software e Serviços de Informática reuniu 37 expositores que apresentaram softwares nacionais e estrangeiros, além de suprimentos e serviços para computação.

Promovida pela terceira vez pela Compucenter e Computer World, apesar de poucos lançamentos, os visitantes tiveram a oportunidade de tomar contato com as principais Empresas de software.

Entre elas, a própria Compucenter que selecionou para esta Feira boas opções como os gerenciadores de projetos Project e Superproject; o Appleworks (programa integrado composto de planilha, editor de texto e Banco de Dados, para a linha Apple IIe) etc.

Os usuários viram ainda: o processador de texto Multimate da Datalogica; o Clipper (compilador do dBase III) da Officer; e o gerenciador de Banco de Dados relacional ZIM da Matrix.

A Cincom Systems também esteve presente neste evento, demonstrando os seguintes softwares: Mantis (linguagem de 4.ª geração para desenvolvimento de aplicações "on-line"); Net/Master (sistema de gerenciamento de Rede); PC Contact para conexão micro-mainframe etc.

Em paralelo à Feira foi organizado um ciclo de 12 palestras sobre os seguintes temas: "Estado da Arte na Linguagem de 4.ª Geração", "Software para Ligação Micro-Mainframe", "Sistemas Gerenciadores de Banco de Dados Relacionais", "Inteligência Artificial e Expert Systems", e outros.

De acordo com o balanço feito pelo diretor da Compucenter, Elias Silmar El-Beck, 60% dos softwares apresentados na III Expo Soft foram estrangeiros e aproximadamente dez mil pessoas compareceram ao evento, ultrapassando assim, as expectativas. Segundo ele, isto está relacionado ao crescimento do mercado brasileiro de software, que no ano passado movimentou US$ 15 milhões e a previsão para 1986 é de US$ 30 milhões. ***M.R.***

O Vale do Silício do Brasil

Está nascendo, no Brasil, a primeira iniciativa conjunta entre universidade, governo e empresários, para a instalação do primeiro Pólo de desenvolvimento de Alta Tecnologia no País.

A criação da ***Ciatec*** - Companhia de Desenvolvimento do Pólo de Alta Tecnologia em Campinas, cidade do interior de São Paulo, lançada oficialmente pelo Presidente José Sarney, em julho passado, inspirou-se em idênticas experiências realizadas em outros países, como Estados Unidos (Vale do Silício), França (Grnoble), Japão (Tsukuba) e Reino Unido (Science Parks).

O plano para a criação de um Pólo industrial gerador de alta tecnologia, principalmente na área de informática, surgiu na década de 70, quando um grupo de pesquisadores da Unicamp (Universidade de Campinas) começou a discutir a implantação na região, de centros avançados de pesquisa. E já no final da década de 70, as discussões se transformaram em um Projeto que contou com o apoio da Prefeitura de Campinas, da Telebrás, do CPFL (Companhia Paulista de Força e Luz) e do CTI (Centro Tecnológico de Informática).

A ***Ciatec*** permitirá a implantação das indústrias numa área delimitada de 248 mil metros localizada ao lado do CTI, além de garantir às mesmas, incentivos fiscais - aprovados pela Prefeitura - como isenção de ISS e IPTU, durante cinco anos. Além disso, o Badesp (Banco de Desenvolvimento do Estado de São Paulo) promoverá planos de financiamentos às empresas que ali se instalarem, com prazos médios de amortização de 12 anos, dois anos de carência, entre outras vantagens.

A escolha de Campinas para a implantação do Pólo Industrial de Tecnologia na área de informática, se deu porque já existem na região, centros avançados de pesquisas como o CTI, a PUCC (PUC Campinas), o CPqD e a própria Unicamp e que irão manter convênios com a ***Ciatec*** e as indústrias, trocando conhecimentos tecnológicos e científicos. ***A.L.A.***

Calendário do mês

15/08 - ***Informática para dirigentes: Conceitos e Estratégias*** - São Paulo - SCI (Sistemas, Computação e Informática) - Tel.: (011) 289-0099.

18/08 - ***Linguagem BASIC sob CP/M*** - Rio de Janeiro - Instituto de Tecnologia ORT - (021) 226-3192.

De 18 a 19/08 - ***DOS - Sistema Operacional para Microcomputadores 16 Bits*** - Belo Horizonte - BMS (Belgo Mineira Sistemas) - Tel.: (031) 226-6644.

De 18 a 24/08 - ***Informática/86 - VI Feira Internacional de Informática*** - Rio de Janeiro - SUCESU.

De 18 a 26/08 - ***Wordstar*** - Belo Horizonte - BMS (Belgo Mineira Sistemas) - Tel.: (031) 226-6644.

De 20 a 22/08 - ***Programando em dBase III*** - São Paulo - SPCI (São Paulo Computer Institute) - Tel.: (011) 883-0355.

De 20 a 22/08 - ***Lotus 1-2-3*** - Belo Horizonte - BMS (Belgo Mineira Sistemas) - Tel.: (031) 226-6644.

De 20 a 22/08 - ***Automação de escritórios, origem, evolução e perspectivas*** - IBPI (Instituto Brasileiro de Pesquisa em Informática) - Rio de Janeiro - Tel.: (021) 286-6891.

De 21 a 22/08 - ***Sistemas de Supervisão e Controle Computadorizado de Energia*** - São Paulo - ABACE (Associação Brasileira de Administração e Conservação de Energia) - Tel.: (011) 287-2627/283-2967.

25/08 - ***MS-DOS Básico*** - São Paulo - Servimec - Tel.: (011) 222-1511.

De 25 a 26/08 - ***Centro de Informações, Implantação e Operação*** - Rio de Janeiro - IBPI (Instituto Brasileiro de Pesquisa em Informática) - Tel.: (021) 286-6891.

De 25 a 27/08 - ***Lotus 1-2-3: Conceitos Básicos*** - Access Information Systems - São Paulo - Tel.: (011) 530-1031.

De 25 a 27/08 - ***Wordstar*** - São Paulo - Servimec - Tel.: (011) 222-1511.

De 25 a 27/08 - ***Programando em dBase II*** - São Paulo - SPCI (São Paulo Computer Institute) - Tel.: (011) 883-0355.

De 25 a 29/08 - ***Aplicativos para o Microcomputador*** - Porto Alegre - Inteligência Treinamento em Informática - Tel.: (0512) 26-1988.

De 25 a 29/08 - ***Análise e Projeto Estruturado para Sistemas On-Line*** - Rio de Janeiro - SCI (Sistemas, Computação e Informática) - Tel.: (021) 294-9292.

De 25 a 29/08 - ***Ligação Micro-Mainframe*** - Porto Alegre - Inteligência Treinamento em Informática - Tel.: (0512) 26-1988.

De 27 a 29/08 - ***Sistemas Distribuídos*** - Rio de Janeiro - IBPI (Instituto Brasileiro de Pesquisa em Informática) - Tel.: (021) 286-6891.

De 27 a 29/08 - ***Os microcomputadores como Ferramentas do Centro de Informática*** - Rio de Janeiro - SCI (Sistemas, Computação e Informática) - Tel.: (021) 294-9292.

De 27 a 29/08 - ***Fundamentos de Bancos de Dados*** - São Paulo - SCI (Sistemas, Computação e Informática) - Tel.: (011) 289-0099.

De 27 a 29/08 - ***dBase II*** - São Paulo - Servimec - Tel.: (011) 222-1511.

De 28 a 29/08 - ***Wordstar*** - Rio de Janeiro - Servimec - Tel.: (021) 221-6067.

De 28 a 29/08 - ***MS-DOS Básico*** - Rio de Janeiro - Servimec - Tel.: (021) 221-6067.

De 01 a 08/10 - ***Logo para educadores e psicólogos*** - Rio de Janeiro - Instituto de Tecnologia ORT - Tel.: (021) 226-3192.

De 02 a 05/09 - ***Banco de Dados em Microcomputador*** - Belo Horizonte - BMS (Belgo Mineira Sistemas) - Tel.: (031) 226-6644.

De 02 a 10/09 - ***Lotus 1-2-3*** - Belo Horizonte - BMS (Belgo Mineira Sistemas) - Tel.: (031) 226-6644.

De 30 a 05/09 - ***Teleprocessamento e Redes de Computação*** - Rio de Janeiro - IBPI (Instituto Brasileiro de Pesquisa em Informática) - Tel.: (021) 286-6891.

De 03 a 05/09 - ***O uso de microcomputador na área de Marketing e Vendas*** - São Paulo - SENAC - Tel.: (011) 256-5522.

De 03 a 05/09 - ***Comunicação para Microcomputadores*** - São Paulo - 3I Informática - Tel.: (011) 521-9509.

De 08 a 12/09 - ***Linguagem "C"*** - Porto Alegre - Inteligência Treinamento em Informática - Tel.: (0512) 26-1988.

De 08 a 14/09 - ***Introdução ao Lotus 1-2-3*** - São Paulo - SENAC - Tel.: (011) 256-5522.

De 08 a 23/09 - ***Assembler Z-80*** - Rio de Janeiro - Instituto de Tecnologia ORT - Tel.: (021) 286-7842.

De 08 a 30/09 - ***Microprocessador I*** - São Bernardo do Campo - IECAT (FEI - Faculdade de Engenharia Industrial) - Tel.: (011) 419-0200.

De 10 a 12/09 - ***Projeto de Circuitos de Alta Integração (LSI e VLSI)*** - Rio de Janeiro - IBPI (Instituto Brasileiro de Pesquisa em Informática) - Tel.: (021) 286-6891.

De 15 a 19/09 - ***O microcomputador no Ensino*** - Porto Alegre - Inteligência Treinamento em Informática - Tel.: (0512) 26-1988.

De 15 a 24/09 - ***BASIC I*** - São Bernardo do Campo - IECAT (FEI - Faculdade de Engenharia Industrial) - Tel.: (011) 419-0200.

16/09 - ***Introdução à Microinformática*** - Rio de Janeiro - Datamicro - Tel.: (021) 511-0395.

17/09 - ***MUMPS para Microcomputadores*** - Rio de Janeiro - Datamicro - Tel.: (021) 511-0395.

De 17 a 19/09 - ***Sistemas Videotexto*** - São Paulo - CKL (Treinamento Empresarial Avançado).

18/09 - ***Gerenciador de Base de Dados*** - Rio de Janeiro - Datamicro - Tel.: (021) 511-0395.

De 18 a 19/09 - ***O microcomputador como ferramenta no processo de tomada de decisões*** - São Paulo - SENAC - Tel.: (011) 255-0066.

De 22 a 29/09 - ***Processadores de Textos - subsídios para escolha*** - São Paulo - SENAC - Tel.: (011) 255-0066.

De 24 a 26/09 - ***Sistemas de Apoio à decisão*** - São Paulo - CKL (Treinamento Empresarial Avançado).

Caro leitor:

Estamos pesquisando a receptividade de nossas seções junto a você. Para isto solicitamos que qualifiquem-nas através de notas e conceitos que vão do Excelente ao Péssimo. Pedimos que nos enviem este encarte, pois através dele poderemos melhorar cada vez mais a nossa MICROHOBBY.

Nesta primeira experiência daremos uma relação apenas das seções e nas edições posteriores ampliaremos a pesquisa, passando a codificar todas as matérias.

Coloque o código da nota/conceito ao lado de cada código da seção. Os códigos referentes às notas, são os seguintes:

A - Para as seções que você considera "Excelente", ou que atenda plenamente a sua necessidade.

B - Para as seções que você considera "Boa", ou que atenda apenas em alguns aspectos o seu interesse.

C - Para as seções que você considera "Razoável", ou que não atenda o seu interesse, mas é uma informação a mais.

D - Para as seções que você considera "Sem valor, ou inútil".

E - Para as seções "Péssimas", abaixo da crítica.

CÓDIGO DA SEÇÃO	NOTA/CONCEITO	CÓDIGO DA SEÇÃO	NOTA/CONCEITO
C. 001		C. 010	
C.U. 002		S.O. 011	
M.P. 003		P. 012	
R.S. 004		V.V. 013	
M.TK-85 005		P.D.A. 014	
E.TK-2000 006		L. 015	
LOG. 007		R.TK-90X 016	
A.A. 008			
H. 009			

MICROHOBBY PRESS

O Cruzado e o conserto de micros

O Plano Cruzado do Governo do Presidente José Sarney está deixando suas influências até mesmo na área de Assistência Técnica para microcomputadores brasileiros. Esta informação foi divulgada pelo diretor comercial da MS Indústria Eletrônica, Sidney Dalben, uma das primeiras empresas a atuar neste segmento no País.

Segundo Sidney Dalben, o usuário voltou a investir um pouco mais na conservação de seu equipamento realizando revisões adequadas e periódicas em seu micro o que, de acordo com Dalben, não acontecia no período dos anos 84 e 85, antes do Cruzado.

Para o diretor da MS, até a decretação do Pacote, o usuário tentava consertar o seu micro por conta própria e acabava acumulando uma série de problemas.

A MS atualmente atende a assistência técnica de várias linhas de equipamentos o que fez com que a empresa ampliasse o seu quadro de funcionários de 45 para 55 profissionais, entre técnicos e engenheiros que já atendem, em média, 1100 atendimentos mensais. ***A.L.A.***

Faturamento recorde da Microtec

Uma das empresas fabricantes do padrão IBM-PC no Brasil, a Microtec Sistemas S/A fechou o primeiro semestre deste ano com um faturamento recorde da ordem de Cz$ 219,9 milhões, com aproximadamente 2 mil e 900 máquinas comercializadas, o que representou um crescimento, em suas vendas, de 184,2 em máquinas.

Segundo pesquisas realizadas pela empresa, os maiores compradores de suas máquinas continuam sendo os estados do Rio de Janeiro e São Paulo o que faz com que a base instalada da empresa alcance o índice de sete mil unidades.

Para o segundo semestre, a Empresa prevê um faturamento bruto de Cz$ 353 milhões, estimando uma venda de 3.700 máquinas. A.L.A.

Foto fornecida pela Prodesp - Elyon Rodrigues Pula

Da esquerda para a direita: Wilson Lazarini, Roseleine Villar D. Furquim (Diretora de produção), Clóvis de Barros Carvlho (Secretário de Planejamento do Estado), Egydio Bianchi (Presidente da Prodesp), Antonio Funari, José Ricca e Renè Lapyda.

Nova diretoria na Prodesp

Com a saída de Wilson Lazzarini, que passou a se dedicar à SUCESU/SP no cargo de presidente da entidade, a Prodesp - Cia. de Processamento de Dados do Estado de São Paulo, passou a ter um novo diretor na sua área de sistemas.

René Lapyda, professor do Instituto de Matemática e Estatística da Universidade de São Paulo e ex-gerente de estudos e projetos especiais da Caixa Econômica do Estado de São Paulo é o novo diretor da área de sistemas.

Ao mesmo tempo tomou posse, também, o novo diretor administrativo-financeiro da entidade, Antonio Funari Filho, que vinha ocupando, até então, a chefia de gabinete da presidência da Prodesp. ***A.L.A.***

Informática no Nordeste

Um Projeto de realização de uma feira de informática nas regiões Norte e Nordeste começa agora a ser concretizado. A Guazzelli Associados, empresa especializada na organização e realização de Feiras nesta área, na cidade de São Paulo, fechou acordo entre os Governos das duas regiões para a realização, no próximo ano (de 9 a 17 de maio), da I Infonor - I Feira Norte/Nordeste de Informática, que irá se realizar no Centro de Convenções de Pernambuco, em Recife.

Num período de nove dias, a I Infonor mostrará os principais lançamentos em computadores, periféricos, acessórios e softwares nos estandes das principais empresas do setor e realizará, em paralelo, o I Encontro Norte/Nordeste de Informática.

O lançamento oficial da I Infonor será feito durante a realização da VI Feira Internacional de Informática, no Rio de Janeiro. ***A.L.A.***

Telex

Uma nova rede nacional de treinamento na área de informática estará sendo implantada a partir do segundo semestre deste ano. O Cetil Processamento de Dados, um ***bureau*** de serviços conceituado no setor, com 18 anos de atuação no mercado e filiais espalhadas pelas principais capitais do País, está ampliando o seu departamento de treinamento, passando a oferecer cursos, para a formação de mão-de-obra nas suas 19 filiais, cinco associadas e mais duas empresas do Grupo.

Para esta ampliação, segundo Sebastião Tavares Pereira, diretor comercial da Empresa, estão sendo investidos cerca de Cz$ 3 milhões e adquiridos 32 micros.

A experiência do projeto de formação de mão-de-obra, que visa atender ao mercado profissional da área de informática, começou nas filiais paulistas e do Paraná do Cetil e, após realizada a avaliação dos resultados, o Grupo decidiu ampliar esta experiência através da realização de 526 cursos, durante o próximo (primeiro) ano de funcionamento, da divisão de treinamento.

RESENHAS DE SOFTWARE

Passo a Passo

Um curso de linguagem BASIC.

Elaine Kelly Lima

A linguagem BASIC foi desenvolvida em 1964 e passou a ser adaptada para diversos equipamentos tornando-se, em pouco tempo, extremamente conhecida devido a duas características que a fazem abranger quase todas as áreas profissionais onde computadores de pequeno porte são muito utilizados: a diversidade de comandos e sua flexibilidade.

Nas escolas de computação, o curso de BASIC é o mais procurado, tanto por quem deseja aprender programação somente como um hobby como também por quem quer usar o aprendido como futura profissão. Sendo este tipo de curso, na maioria das vezes, o primeiro contato do aluno com programação, existe a necessidade de um preparo de aulas introdutórias que transmitam ao novato as primeiras noções e técnicas para desenvolvimento de programas.

Percebendo a necessidade de formar programadores com bases técnicas mais sólidas, a ***Potencial Software*** desenvolveu um curso de BASIC, versão Applesoft para iniciantes, onde o micro é o professor. O curso é composto por três partes:

- um software que ilustra o aprendizado, apresentando ao usuário o processamento causado por cada instrução ou comando;
- um tutorial que complementa o software e explica detalhadamente à cada nova tela, o que está sendo exibido e qual deve ser o procedimento do usuário. Ao mesmo tempo que funciona tal qual um manual de BASIC, também serve como manual do software;
- um caderno de problemas práticos que auxiliarão o aluno na assimilação de cada nova matéria apresentada pelo tutorial e software.

O curso é constituído por dez aulas e um teste final que inclui exercícios referentes a todo seu conteúdo. Cada aula é dividida em três fases: ***parte 1,*** que é uma introdução a determinado assunto; ***parte 2,*** que é um complemento da primeira etapa; e um ***teste*** cujos exercícios propostos são referentes ao aprendido nas aulas já estudadas. É através da última etapa que o usuário será avaliado e ficará sabendo se está apto, ou não, a passar para a aula seguinte.

Todas as aulas devem ser acompanhadas ao mesmo tempo com o software e o tutorial. Durante a exibição da aula no monitor do seu Apple, você deverá prestar muita atenção, pois em determinados momentos será necessário que a tecla RETURN seja pressionada, isso ocorre para que o aluno tenha o tempo que achar necessário para examinar os visuais, ler o tutorial e entender o assunto.

Parte a parte

Como cada parte do curso apresenta assuntos distintos, resolvemos analisar aqui todas as aulas, parte por parte, para que o leitor tenha uma idéia exata do conteúdo abrangido pelo curso.

Aula 1 - parte 1, aqui o aluno será apresentado ao equipamento com o qual está trabalhando. Através de um esquema especial é exibido no monitor a memória e a utilidade do interpretador; o que é a Linguagem de Máquina; algumas palavras reservadas do BASIC; e a instrução PRINT, que é responsável pela exibição de constantes e conteúdo de variáveis em tela.

Em conjunto são propostos alguns problemas que permitem ao usuário conhecer os principais erros de sintaxe que podem ocorrer ao usarmos errôneamente, esta instrução.

Aula 1 - parte 2, nesta fase o usuário aprenderá a utilizar a instrução PRINT juntamente com os operadores aritméticos, para exibir resultados de cálculos de expressões matemáticas numa única instrução. Para aproveitar o embalo, também é demonstrado, de maneira interessante, a ordem de importância dos operadores aritméticos.

Os exercícios desta aula serão facilmente solucionados se o usuário tiver aprendido a lição.

Aula 2 - parte 1, o aluno recebe conceitos sobre o que são comandos diretos, o que é um programa e qual a diferença entre eles. E o por que de deixarmos um intervalo entre as linhas de um programa.

Aula 2 - parte 2, aqui serão distinguidos comandos e instruções e depois ensina-se a utilizar dois comandos, um para apagar programa da memória e outro para apagar apenas blocos. Em seguida, o aluno aprende a parar o processamento de um programa, através de duas instruções diferentes.

Aula 3 - parte 1, nesta etapa será ensinado o que é uma variável, como elas podem ser nomeadas e quando são usadas. Apresenta-se também a instrução que possibilita jogar o valor de uma expressão para uma variável. Os problemas propostos são cinco programas simples que usam variáveis e expressões para que o aluno analise e compreenda.

Aula 3 - parte 2, ensina a instrução que permite a entrada de dados, via teclado, em variáveis durante a execução do programa. Como exercícios são propostos alguns temas para que você comece a desenvolver programinhas simples e também pede-se que o aluno execute três programas que lhe mostrarão, na prática, como o aprendido até o momento é processado.

Aula 4 - parte 1, aqui uma das estruturas de controle do BASIC é ilustrada e muito bem explicada, o IF...THEN..., além disso, também ensina-se a diferença entre desvio condicional e incondicional para que o aluno comece a criar estruturas para os seus programas.

Os exercícios desta fase mostram os dois tipos de desvios e o que vem a ser um loop.

Aula 4 - parte 2, é um complemento para a primeira parte, e ensina como usar os operadores relacionais nos desvios incondicionais. Para praticar, o caderno de exercícios apresenta dois programas com todo conteúdo ensinado até esta etapa do curso. O aluno deverá digitar, executar e analisar o resultado obtido.

Aula 5 - parte 1, o aluno aprende uma técnica simples, porém muito utilizada em programação, ***como e porque usar os famosos "contadores".*** Os exercícios práticos apresentam programas maiores que os vistos até agora e se o aluno desejar realmente aprender, deverá executá-los e assim, analisar seus processamentos.

Aula 5 - parte 2, aqui é feito um resumo das funções apresentadas por esta versão do BASIC, e também é explicado detalhadamente o processamento realizado por cada uma delas na memória do micro. Nos exercícios propostos aprende-se a contar de 5 em 5 por um meio mais complicado. Além disso, com dois exemplos muito interessantes o aluno passará a entender a função que gera números aleatórios.

Aula 6 - parte 1, a explicação a respeito da instrução PRINT é complementada e o aluno passa a conhecer certos macetes para a formatação da tela do vídeo.

Lição 6 - parte 2, o armazenamento de variáveis numéricas e alfanuméricas é

ilustrado detalhadamente, tornando fácil ao aluno assimilar este processamento interno da máquina.

Lição 7 - parte 1, mostra detalhadamente como deve-se utilizar as instruções para chamar uma subrotina e causar um retorno . Desta maneira, o usuário percebe que não é difícil começar a criar estruturas para tornar os programas tão confusos de se entender por causa do uso excessivo de GOTO's.

Lição 7 - parte 2, o aluno aprende uma nova maneira de armazenar constantes em variáveis durante a execução do programa através dos comandos READ...DATA. A partir daqui, os problemas propostos são programas que necessitam de alterações para haver uma mudança positiva no processamento.

Lição 8 - parte 1, outra estrutura de controle muitíssimo importante e que todas as versões BASIC possuem é apresentada ao aluno nesta lição, o FOR...NEXT.

Lição 8 - parte 2, quando chegar aqui, se o aluno não souber o que é uma tabela vetorial, bidimensional ou tridimensional, deverá ler atentamente o tutorial e prestar bastante atenção às explicações do software. O segundo ilustra também como são alocadas na memória estes tipos de variáveis.

Lição 9 - parte 1, se o aluno já chegou a esta lição é porque ele consegue desenvolver muitos programas e certamente está habituado aos comandos e instruções BASIC. Nesta lição é ensinado como podemos criar cadeias de FOR...NEXT's sem causar erros de sintaxe, nem erros de lógica.

Lição 9 - parte 2, é muito ilustrativa e apresenta ao aluno a tela de baixa resolução gráfica e os comandos para desenvolver nela alguma coisa. A apresentação dada pelo software é muito gostosa de assistir, porém se o aluno não tiver lido o tutorial ficará um tanto quanto perdido para responder aos exercícios.

Lição 10 - parte 1, pra complementar este curso de introdução são apresentados alguns caracteres especiais utilizados para facilitar a vida dos programadores.

Lição 10 - parte 2, como esta é a última etapa é feito um resumo sobre o processamento interno na exponenciação, ordem aritmética e notação científica.

Além disso, você deverá fazer o teste final que contém tudo o que foi dado, sendo que através dele o aluno poderá se auto avaliar.

No final do tutorial há uma relação de todas as palavras reservadas desta versão de BASIC, seus formatos e utilidades. ◇

Banco de Dados

A Special Systems oferece ao mercado mais uma opção para os usuários dos micros compatíveis ao TK-90X. O programa Banco de Dados possibilita ao usuário o acúmulo de informações e sua rápida recuperação.

Seu desenvolvimento foi feito em Assembler e permite a formatação das fichas do arquivo, ou seja, ele molda a forma com que os dados serão recebidos, possibilitando também, a inclusão, exclusão, alteração, manipulação de uma única ficha, leitura de registros, ordenação por qualquer campo, cópia de uma ficha na impressora, leitura e gravação de um Banco de Dados em fita, além de outras vantagens como a opção informação que indica ao usuário, a quantidade de memória disponível para inclusão de fichas.

A auto-explicação do programa possibilita seu fácil entendimento e a mais rápida familiarização com o sistema. A fita vem acompanhada por um folheto de explicação dos principais comandos do Banco de Dados.

Banco de Dados está sendo comercializado ao preço de Cz$ 100,00 e pode ser conseguido na Stop - Praia do Icaraí, 211 - Loja 3 - Rio de Janeiro. ***P.P.A.*** ◇

ZZOOM

A sensação de voar dentro de um dos aviões de combate, na Segunda Guerra Mundial devia ser fantástica. Agora, imagine-se dentro de um destes aviões sendo o piloto e tendo de salvar a vida de centenas de pessoas que fogem de bombardeios, submarinos e tanques de guerra.

Pois bem, a Multisoft lança no mercado um novo software de 48K de RAM para os micros compatíveis ao TK-90X, que simula exatamente o que foi dito e muito mais.

As telas de jogo são muito bem definidas e possibilitam ao usuário a sensação quase que real, de tudo que acontece no vídeo.

O jogo se inicia com o piloto automático, que posiciona o avião em sua posição de combate. Logo após, o painel de controle que se localiza no canto superior da tela, informa a condição que você se encontra no momento:

- ***condição verde:*** o piloto automático é desligado e você assume o comando;
- ***condição amarela:*** é o alerta de combate;
- ***condição vermelha:*** é o Ataque, nesta condição o usuário deverá destruir os aviões, tanques e submarinos para evitar ser atingido pelos mísseis.

O programa é acompanhado por um folheto explicativo, contendo as teclas utilizadas no jogo e como as utilizar. O jogo é rápido e no mesmo nível do software ***Air Wolf.***

ZZOOM pode ser encontrado em todos os revendedores Microdigital, ao preço de Cz$ 59,50. ◇

RESENHAS DE SOFTWARE

R.S. 004

TKdrez

Desenvolvido para o TK-85 pela Microdigital Eletrônica sob a marca Microsoft, este software é o adversário perfeito para qualquer enxadrista, iniciante ou mestre internacional (com duas versões: I e II), isto porque possui sete níveis de dificuldade sendo que, para cada um, o tempo que o programa demora para efetuar uma jogada é diferente.

No nível mais fácil, esse tempo é de aproximadamente dois segundos. Se o usuário jogá-lo razoavelmente bem, o programa realiza sua jogada em até 15 segundos. Se o nível escolhido for à altura dos jogadores com prática, o programa demora até 40 segundos.

Caso o usuário considere-se bom jogador, o nível de dificuldade aconselhável é o 3, cuja jogada do programa demora até três minutos; já os jogadores bem experientes devem selecionar o nível 4 cuja demora é de cinco minutos. Porém, se o usuário for um mestre internacional sua escolha deverá ser feita entre os níveis de dificuldade 5 ou 6, onde o padrão de jogo será excelente, mas o tempo que temos de esperar pelo lance do programa é muitíssimo grande.

No tabuleiro, as peças brancas são exibidas sobre fundo preto, e as pretas sobre fundo branco. O tabuleiro é uma matriz dimensionada verticalmente de 1 até 8 e horizontalmente, de A até H. As peças são indicadas pelas seguintes letras: R - rei; D - dama; T - torre; B - bispo; C - cavalo e P - peão.

A partida termina somente quando ocorre o xeque-mate e quando isso acontecer, será necessário pressionar qualquer tecla para que o programa retorne ao menu principal.

Além da opção de jogo, são oferecidas mais duas: analisar ou carregar tabuleiro do jogo anterior, que não tenha acabado e que, por este motivo, foi armazenado em fita.

Na opção ***analisar***, o usuário pode montar um novo tabuleiro ou modificar o que está na memória. Atavés dela o programa encontra o próximo lance da partida.

Se o usuário possuir uma TK Printer, será possível copiar o tabuleiro a qualquer momento.

Não perca este desafio e enfrente o TKdrez.

O programa está custando Cz$ 48,50 a versão I e Cz$ 51,50 a versão II e pode ser encontrado nos revendedores da empresa ou em suas unidades de assistência técnica. *E.K.L.* ◇

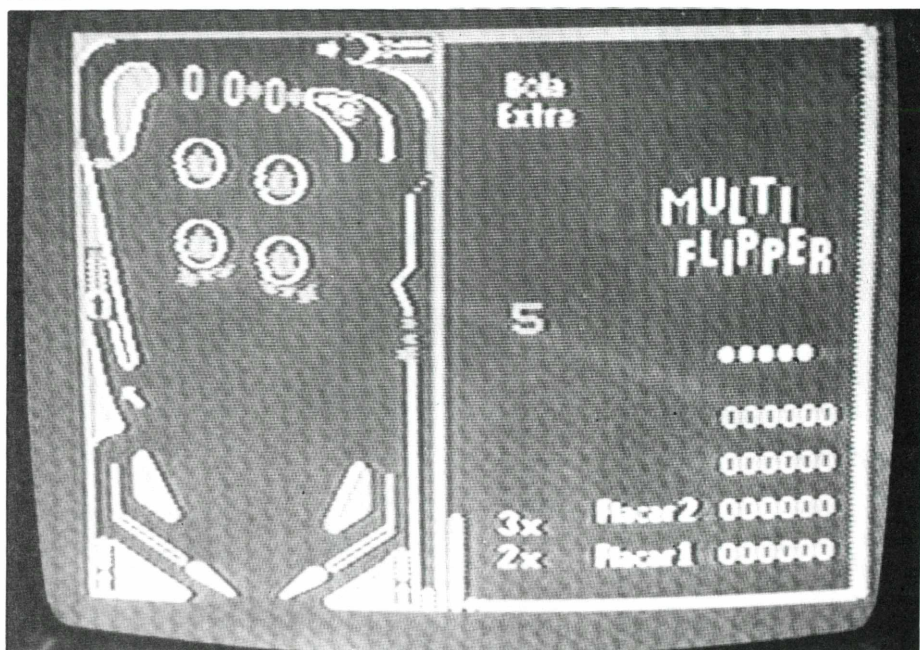
Air Wolf

Air Wolf é um produto da Multisoft para os compatíveis do TK-90X (48K), que possui uma ótima resolução gráfica e oferece muitos recursos sonoros, além de forçar o usuário a utilizar toda sua perícia para cumprir o objetivo do jogo.

O objetivo do jogo é de ultrapassar obstáculos sem os atingir com o helicóptero. Alguns destes terão de ser destruídos, para desobstruir a passagem do helicóptero. Ao se encontrar um dispositivo em forma de "X", deve-se atingí-lo, utilizando o canhão do helicóptero. Dessa forma, uma nova passagem secreta será aberta e o jogo continua.

Se você gosta de jogos difíceis, este software é ideal para você. As teclas de comando são apresentadas na tela inicial e no pequeno folheto que acompanha a fita cassete.

Outro importante detalhe é que o programa é totalmente desenvolvido em Linguagem de Máquina oferecendo um jogo rápido e divertido.

O programa está custando Cz$ 59,50 e pode ser encontrado nos revendedores e unidades técnicas da Microdigital. ***P.P.A.***

Israel Teixeira

Flipperama

O jogo do futuro para o TK-2000.

A máquina de jogos do futuro está desafiando você para uma partida cheia de emoções. Perícia, habilidade e reflexo são os principais requisitos para o jogador.

Este programa oferece um completo domínio sobre o jogo, transformando o seu equipamento em um sofisticado flipperama, com uma superior capacidade de entretenimento para você e seus amigos.

Este jogo, mundialmente conhecido, chega agora para o micro trazendo assim, a máquina de flipperama para dentro de sua casa.

Todas as características de uma máquina de flipperama foram inseridas neste programa, para que você sinta as mesmas emoções.

O jogo é composto de quatro placares onde cada jogador tem o direito de jogar cinco bolinhas, bônus, canaletas laterais e uma central, dois flippers e a possibilidade de prender as bolinhas durante o jogo etc.

Antes de iniciar a partida, você deve escolher em qual nível deseja jogar.

Os flippers podem ser controlados através do teclado, ou pelo joystick.

Os efeitos sonoros são muito parecidos com os das máquinas de flipperama, dando a sensação de estar diante de uma.

Este programa desenvolvido pela Microsoft para o TK-2000, pode ser encontrado nos revendedores da Microdigital, ao preço de Cz$ 63,50. ◇

MEGA TK-85 - ÚLTIMA PARTE M.TK-85 005

O Cérebro do Mega TK

Paulo Sergio Naddeo D. Lopes
Ricardo Urbini

As 30 rotinas apresentadas nas edições anteriores, possuem cada uma sua função específica no jogo Mega TK. Porém, elas devem ser executadas no momento certo. O jogo também possui muitas variáveis de controle como velocidade, número de tiros dos modelos, valor de cada modelo destruído, que deverão ser ajustadas durante o jogo. O programa principal, além de cuidar disso tudo deve decidir qual caminho tomar para cada finalização do jogo. Portanto, nada melhor do que apelidá-lo de ***Cérebro.***

Na figura 1 temos o seu ***fluxograma.*** Os círculos numerados significam que o ***Cérebro*** está chamando a rotina especificada pelo número. Ao término da execução da rotina chamada, segue-se o programa principal.

Análise do Cérebro

Inicialmente chamamos a rotina SLOW do Sistema Operacional. Se o micro trabalhasse no modo FAST, não veríamos o jogo no vídeo, pois neste modo o TK só faz o REFRESH do vídeo quando pára, na espera do teclado.

A seguir, limpamos a tela e a formatamos para 26 linhas, ou seja, sem linha de status, zerando a variável DFSZ do sistema operacional.

As variáveis

A cada reínicio do jogo carregamos os valores iniciais de suas variáveis. Para isso, transferimos os bytes da área:

40DC - 40F1 (Hexa) para a área 4082 - 4097 (Hexa),
16604 - 16625 16514 a 16535

que dá um total de 20 variáveis, sendo que duas possuem 2 bytes, ***(veja Mega-TK Parte I).*** Tanto o jogador inexperiente como o viciado podem alterar alguns parâmetros para deixar o jogo mais fácil ou mais difícil. Basta alterar o valor inicial da variável certa, na área de 40DC a 40F1 (Hexa). As possíveis alterações são:

Variável VIDAS (16609) - Número inicial de vidas
Valor mínimo: 01.
Valor máximo: 06 (limitação da rotina 17).
Valor atual: 03.

Variável VALMOD (16613) - Valor inicial do modelo (Dividido por 10).
Valor mínimo: 01.
Valor máximo: 05 (Limitação do placar).
Valor atual: 01.

Variável TIROS (16614) - Número de ciclos necessários para gerar os tiros dos modelos (cada ciclo é um passo dos modelos na Tela).
Valor mínimo: 00 (gera tiro a cada ciclo).
Valor máximo: FF (limitação do byte).
Valor atual: 10 (gera tiro a cada 10 ciclos).

Variáveis TIRMOD (16617, 16618, 16619) - Definem o número de modelos que devem gerar os tiros. Cada bit em 1 (entre o bit 0 e o bit 4) de cada TIRMOD é um modelo. À cada geração dos tiros, estas variáveis sofrem uma rotação nos seus bits.
Valor mínimo: 00 (não geram tiro).
Valor máximo: FF (todos geram tiro).
Valor atual: TIRMOD1: A2
TIRMOD2: 15
TIRMOD3: A0

Variável DELAY (16624) - DELAY de jogo.
Valor mínimo: 01.
Valor máximo: FF (limitação do byte).
Valor atual: 06.

O restante das variáveis são de controle do jogo como a posição de tiro do canhão, contadores de ciclos para geração dos tiros dos modelos (FIRE) e diminuição da energia (voltas), posição do canhão, tiro de modelo etc.

A lógica nos movimentos

Esta é a parte mais complicada do jogo: ***qual seqüência de movimentos devemos seguir para que tenhamos a impressão de simultaneidade?***

Pedimos que o leitor acompanhe pelo fluxograma, para facilitar o entendimento.

Primeiramente, lemos o teclado. Se as teclas 5, 8 ou 0 foram pressionadas, executamos a sua função, ou seja, movimentando o canhão uma coluna para a esquerda, movimentando o canhão uma coluna para a direita, ou gerando o tiro do canhão, respectivamente.

Pressionadas ou não algumas destas teclas, devemos apagar o tiro do canhão (se existir) para que este não entre no movimento dos modelos. Em seguida, verificamos qual é o modelo para chamarmos a rotina de movimentação do modelo apropriado. Se forem os modelos 0 ou 1, chamamos a rotina de SCROLL para a direita (esta só faz o SCROLL das 8 primeiras linhas da tela), porém se forem os modelos 2 ou 3 devemos apagar o canhão da tela antes de chamarmos a rotina de movimentação do modelo, para que este não participe do movimento. Estes modelos não possuem tiros, portanto podemos restaurar o canhão após o movimento deles; os modelos 0 e 1 possuem tiros. Assim, logo após terem sido feitos os seus movimentos devemos usar os seus tiros. Esta movimentação é feita em duas partes: a primeira é a movimentação daqueles tiros que estavam nas 8 primeiras linhas da tela que entraram no SCROLL para a direita. Nesta movimentação, eles são ajustados para a coluna certa, uma linha abaixo. Aqueles que passaram para a nona linha entrarão na segunda parte, juntamente com os outros, que é o SCROLL para baixo dos tiros (da linha 8 à linha 21). Só após efetuar esses movimentos é que o canhão será restaurado. Cabe à essa rotina de restauração (22) verificar se um tiro do modelo invadiu as suas posições. Se invadiu, o canhão explode; caso contrário, reacendemos o tiro do canhão que foi apagado para não entrar em nenhum destes movimentos. Cabe à rotina 16 - "ACENDE O TIRO DO CANHÃO" verificar se na posição não tem algum modelo. Este tiro reacende-se duas linhas acima da posição anterior para que seja mais rápido que os tiros dos modelos.

Após acertar as variáveis necessárias completamos um ciclo, voltando à leitura do teclado. Para que um modelo dos tiros 0 ou 1 atravesse toda a tela são necessários 32 ciclos (32 colunas).

Se a variável TIROS valer 10 (hexa), cada modelo poderá soltar dois tiros enquanto cruza toda a tela, pois esta variável diz o número de ciclos necessários para gerar os tiros dos modelos.

Para rodar o jogo basta digitar RAND USR 18057.

```
                 ;**************************************************
                 ;      C E R E B R O  -  PROGRAMA  PRINCIPAL
                 ;**************************************************

4689   CD 0F2B   inicio: call slow        ;coloca o micro no modo SLOW
468C   CD 0A2A   cer010: call cls         ;limpa a tela do micro
```

```
46BF    AF                  xor a                  ;tira todas as linhas de edicao da tela,...
4690    32 4022             ld (dfsz),a            ;...deixando a tela oda para o jogo
4693    21 40DC             ld hl,valini           ;transfere os valores iniciais das variaveis
4696    11 4082             ld de,direcao
4699    01 0016             ld bc,22               ;22 variaveis de programa
469C    ED B0               ldir
469E    CD 414C             call rot03             ;desenha o placar e as vidas
46A1    21 4095     cer020: ld hl,numod
46A4    36 0F               ld (hl),0fh            ;acerta o numero de modelos em 15
46A6    21 408D             ld hl,linscroll        ;acerta o numero de linhas do scroll dos...
46A9    36 08               ld (hl),8              ;...modelos para direita em 8
46AB    21 408F             ld hl,tirmd1           ;acerta os bytes de geracao de tiro dos modelos
46AE    3A 40E9             ld a,(valini+13)       ;pega o valor inicial de tirmod1
46B1    77                  ld (hl),a              ;modelos da terceira linha (tirmd1): 3 tiros
46B2    23                  inc hl                 ;(numero de tiros = numero de bits ON)
46B3    3A 40EA             ld a,(valini+14)       ;pega o valor inicial de tirmod2
46B6    77                  ld (hl),a              ;modelos da segunda linha (tirmd2): 2 tiros
46B7    23                  inc hl
46B8    3A 40EB             ld a,(valini+15)       ;pega o valor inicial de tirmod3
46BB    77                  ld (hl),a              ;modelos da primeira linha (tirmd3): 3 tiros
46BC    CD 40F2             call rot01             ;desenha os modelos 0 e 1
46BF    21 4084     cer030: ld hl,posenergia
46C2    36 1E               ld (hl),1eh            ;acerta a posicao inicial da energia
46C4    21 4089             ld hl,fltiro
46C7    36 00               ld (hl),0              ;inicializa flag de existencia de tiro do canhao
46C9    23                  inc hl
46CA    36 0E               ld (hl),0eh            ;acerta a posicao inicial do canhao (coluna 14)
46CC    CD 411F             call rot02             ;desenha a energia e o canhao
46CF    CD 02BB     cer040: call scankey           ;le teclado
46D2    2C                  inc l                  ;verifica se leu alguma tecla
46D3    28 0C               jr z,cer050            ;se nao leu, desvia
46D5    2D                  dec l
46D6    44                  ld b,h
46D7    4D                  ld c,l
46D8    CD 07BD             call achr              ;determina o codigo da tecla
46DB    7E                  ld a,(hl)
46DC    FE 1C               cp 1ch                 ;verifica se o codigo e 1ch: tecla "0"
46DE    CC 458C             call z,rot25           ;se for tecla 0, chama a rotina que gera o..
46E1    CD 4618     cer050: call rot28             ;...tiro do canhao; sendo ou nao, chama tam-...
46E4    21 4095             ld hl,numod            ;...bem, a rotina que move o tiro do canhao
46E7    AF                  xor a
46E8    BE                  cp (hl)                ;verifica o numero de modelos restantes na tela
46E9    CA 4771             jp z,cer160            ;se for zero, desvia
46EC    21 4088             ld hl,modelo
46EF    7E                  ld a,(hl)
46F0    E6 03               and 03
46F2    FE 02               cp 2                   ;verifica se sao modelos 0,1 ou modelos 2,3
46F4    30 1C               jr nc,cer080           ;se for 2,3 desvia, pois estes nao soltam tiros
46F6    21 4092             ld hl,fire
46F9    3A 408C             ld a,(tiros)
46FC    BE                  cp (hl)
```

```
46FD   30 07              jr nc,cer070       ;se fire > tiros  :  faz-se...
46FF   36 FF              ld (hl),0ffh       ;...fire = 0ffh e gera os tiros
4701   E5                 push hl            ;(fire determina quando gerar os tiros)
4702   CD 44FF            call rot24         ;gera os tiros dos modelos
4705   E1                 pop hl
4706   34         cer070: inc (hl)           ;se fire <= tiros  :  incrementa-se fire
4707   CD 41CA            call rot07         ;roda os modelos para direita
470A   CD 445E            call rot20         ;ajusta os tiros dos modelos
470D   CD 41AF            call rot06         ;faz um scroll dos tiros dos modelos para baixo
4710   18 12              jr cer100          ;(neste scroll o canhao some da tela)
4712   CD 4482    cer080: call rot21         ;apaga o canhao
4715   21 4088            ld hl,modelo
4718   CB 46              bit 0,(hl)         ;verifica se e modelo 2 ou modelo 3
471A   28 05              jr z,cer090        ;se for modelo 2, desvia
471C   CD 4276            call rot12         ;movimenta um passo o modelo 3
471F   18 03              jr cer100
4721   CD 423A    cer090: call rot11         ;movimenta um passo o modelo 2
4724   CD 4499    cer100: call rot22         ;restaura o canhao
4727   FE FF              cp 0ffh            ;se "A" voltar com 0ffh - o canhao nao foi...
4729   20 3B              jr nz,cer150       ;...atingido, desviando para cer150
472B   3E 1B              ld a,1bh
472D   BE                 cp (hl)            ;verifica se foi atingido por um tiro do modelo.
472E   28 03              jr z,cer110
4730   CD 436D            call rot17         ;se foi atingido por um modelo - ajusta o placar
4733   CD 44BE    cer110: call rot23         ;explode o canhao
4736   21 4087    cer120: ld hl,vidas
4739   35                 dec (hl)           ;decrementa o numero de vidas
473A   AF                 xor a
473B   BE                 cp (hl)            ;verifica se terminaram as vidas
473C   38 1C              jr c,cer140        ;se ainda nao terminaram, desvia
473E   CD 45BC            call rot26         ;joga mensagem de fim de partida
4741   CD 02BB    cer130: call scankey       ;le o teclado
4744   2C                 inc l
4745   28 FA              jr z,cer130        ;espera uma tecla
4747   2D                 dec l
4748   44                 ld b,h
4749   4D                 ld c,l
474A   CD 07BD            call achr          ;pega o codigo da tecla lida
474D   7E                 ld a,(hl)
474E   FE 38              cp 38h             ;verifica se foi teclado 'S'
4750   CA 468C            jp z,cer010        ;se sim, recomeca o jogo
4753   FE 33              cp 33h             ;verifica se foi teclado 'N'
4755   20 EA              jr nz,cer130       ;nao sendo 'N'nem S', le teclado novamente
4757   CD 0CDC            call stop          ;se teclou 'N', para o jogo, retornando ao BASIC
475A   CD 43BA    cer140: call rot18         ;ajusta as vidas
475D   CD 4662            call rot29         ;delay de jogo
4760   CD 45DB            call rot27         ;apaga os tiros do modelo e do canhao
4763   C3 46BF            jp cer030          ;retorna para continuar o jogo com novo canhao
4766   CD 4343    cer150: call rot16         ;acende o tiro do canhao
4769   21 4095            ld hl,numod
476C   AF                 xor a
476D   BE                 cp (hl)            ;verifica se ainda tem modelo na tela
```

```
476E   C2 4807            jp nz,cer250        ;se ainda tem, desvia
4771   21 4088   cer160:  ld hl,modelo
4774   34                 inc (hl)            ;passa para o proximo modelo
4775   7E                 ld a,(hl)
4776   E6 03              and 03
4778   FE 01              cp 1                ;verifica se e modelo 0
477A   30 41              jr nc,cer210        ;sendo modelo 1,2 ou 3 : desvia sem alterar os...
477C   21 4085            ld hl,codmod        ;...parametros do jogo. (nao completou o ciclo).
477F   36 84              ld (hl),84h         ;coloca os codigos du modelo 0 na variavel
4781   23                 inc hl
4782   36 07              ld (hl),7
4784   21 4096            ld hl,delay1
4787   CB 3E              srl (hl)            ;o delay1 cai pela metade
4789   AF                 xor a
478A   BE                 cp (hl)
478B   38 02              jr c,cer170
478D   36 01              ld (hl),1           ;nao deixa delay1 = 0
478F   21 408B   cer170:  ld hl,valmod
4792   7E                 ld a,(hl)
4793   06 01              ld b,1
4795   80                 add a,b             ;incrementa o valor do modelo
4796   FE 06              cp 6                ;nao deixa este ser maior que 6
4798   30 01              jr nc,cer180
479A   77                 ld (hl),a           ;atualiza a variavel valmod
479B   21 408C   cer180:  ld hl,tiros
479E   3E 04              ld a,4
47A0   BE                 cp (hl)             ;verifica se tiros esta com seu valor minimo ...
47A1   28 03              jr z,cer190         ;...(quanto menor o seu valor mais tiros os mo...
47A3   35                 dec (hl)            ;..delos soltam). Se nao estiver, decrementa de 1
47A4   18 0B              jr cer200
47A6   21 4097   cer190:  ld hl,delay2
47A9   3E 03              ld a,3
47AB   BE                 cp (hl)
47AC   30 03              jr nc,cer200        ;nao deixa delay2 < 3
47AE   35                 dec (hl)            ;decrementa de 3 unidades o delay2
47AF   35                 dec (hl)
47B0   35                 dec (hl)
47B1   CD 4662   cer200:  call rot29          ;delay de jogo
47B4   CD 4482            call rot21          ;apaga o canhao
47B7   CD 45DB            call rot27          ;apaga os tiros do modelo e do canhao
47BA   C3 46A1            jp cer020           ;retorna p/continuar o jogo com novos parametros
47BD   FE 02     cer210:  cp 2                ;verifica se e modelo 1
47BF   30 0A              jr nc,cer220        ;sendo modelo 2 ou 3, desvia
47C1   21 4085            ld hl,codmod
47C4   36 06              ld (hl),6           ;coloca os codigos do modelo 1 na variavel
47C6   23                 inc hl
47C7   36 86              ld (hl),86h
47C9   18 E6              jr cer200
47CB   FE 03     cer220:  cp 3                ;verifica se e modelo 2
47CD   30 0F              jr nc,cer230        ;sendo modelo 3, desvia
47CF   21 4085            ld hl,codmod
```

```
47D2   36 09              ld (hl),9          ;coloca o codigo do modelo 2 na variavel
47D4   23                 inc hl
47D5   36 00              ld (hl),0
47D7   21 4082            ld hl,direcao      ;acerta o valor inicial da direcao de seu mo...
47DA   36 FD              ld (hl),0fdh       ;...vimento
47DC   18 10              jr cer240
47DE   21 4085    cer230: ld hl,codmod
47E1   36 89              ld (hl),89h        ;coloca o codigo do modelo 3 na variavel
47E3   23                 inc hl
47E4   36 00              ld (hl),0
47E6   21 4082            ld hl,direcao      ;acerta o valor inicial da direcao e do passo...
47E9   36 00              ld (hl),0          ;...do seu movimento
47EB   23                 inc hl
47EC   36 03              ld (hl),3
47EE   CD 4662    cer240: call rot29         ;delay de jogo
47F1   CD 4482            call rot21         ;apaga o canhao
47F4   CD 45DB            call rot27         ;apaga os tiros do modelo e do canhao
47F7   CD 416D            call rot04         ;desenha os modelos (do tipo 2 ou 3)
47FA   21 4095            ld hl,numod
47FD   36 0A              ld (hl),0ah        ;inicializa a variavel de numero de modelos=10
47FF   21 41CB            ld hl,rot07+1      ;acerta o numero de linhas que irao rodar...
4802   36 15              ld (hl),15h        ;...para direita na rotina 07
4804   C3 46BF            jp cer030          ;retorna para continuar o jogo com novo modelo
4807   21 408E    cer250: ld hl,voltas
480A   34                 inc (hl)           ;incrementa o contador de passos do modelo
480B   7E                 ld a,(hl)
480C   FE 20              cp 20h             ;verifica se o modelo ja deu 20 passos na tela
480E   38 0F              jr c,cer260        ;se ainda nao, desvia
4810   CD 4325            call rot15         ;caso contrario, decrementa a energia
4813   21 4084            ld hl,posenergia
4816   AF                 xor a
4817   BE                 cp (hl)            ;verifica se acabou a energia
4818   28 0B              jr z,cer270        ;se sim, desvia
481A   21 408E            ld hl,voltas
481D   36 00              ld (hl),0          ;caso contrario, zera o contador
481F   CD 4662    cer260: call rot29         ;delay de jogo
4822   C3 46CF            jp cer040          ;retorna para proximo passo
4825   06 05      cer270: ld b,5             ;terminando a energia, perde o canhao
4827   CD 41A7            call rot05         ;delay auxiliar
482A   21 408E            ld hl,voltas
482D   36 00              ld (hl),0          ;zera o contador
482F   06 04              ld b,4
4831   C5         cer280: push bc
4832   CD 466E            call rot30         ;inverte o video
4835   06 05              ld b,5
4837   CD 41A7            call rot05         ;delay auxiliar
483A   C1                 pop bc
483B   10 F4              djnz cer280
483D   CD 4482            call rot21         ;apaga o canhao (perde o canhao sem explodir)
4840   C3 4736            jp cer120
```

▷

Fluxograma
INÍCIO
SLOW
CLS
DFSZ = ∅
TRANSF. DE VARIÁVEIS
IMPRESSÃO DO PLACAR E DAS VIDAS
3
NUMOD = ∅F
LINSCROLL = ∅8
PEGA VALORES DOS TIRMODS NA ÁREA INICIAL
IMPRESSÃO DOS MODELOS 1 e 2
1
POSENERGIA = 1E
FLTIRO = ∅∅
POSCAN = ∅E
IMPRESSÃO DA ENERGIA E DO CANHÃO
2
TECLA 5, 8 ou NÃO
MOVIMENTA O CANHÃO P/ ESQUERDA
13
MOVIMENTA O CANHÃO P/ DIREITA
14
TECLA ∅?
NÃO
SIM
GERAÇÃO DO TIRO DO CANHÃO
25
APAGA O TIRO DO CANHÃO
28
NUMOD : ∅
MODELO 1,2 ou 3,4
FIRE : TIROS
FIRE : FF
INCR. FIRE
GERAÇÃO DO TIRO DO MODELO
24
APAGA O CANHÃO
21
MODELO 3 ou 4
MOV. DO MOD. 4
12
MOV. DO MOD. 3
11
SCROLL P/DIREITA
7
MANIPULAÇÃO DO TIRO DO MODELO
2∅
SCROLL DOS TIROS
6
RESTAURAÇÃO DO CANHÃO
22
A : 'FF'
A = "."
A : (HL)
AJUSTE DO PLACAR
17
EXPLOSÃO DO CANHÃO
23
DECREM VIDAS
VIDAS : ∅
IMPRESSÃO DAS VIDAS
18
DELAY DE JOGO
29
APAGA OS TIROS
27
MENSAGEM DE FIM DE PARTIDA
26
TECLA S ou N
FIM
B : ∅
DELAY
5
INVERSÃO DO VÍDEO
3∅
B = ∅4
VOLTAS = ∅
APAGA OS TIROS
APAGA O CANHÃO
DELAY DE JOGO
SCROLL = = 15
NUMOD = = ∅A
IMPR. DOS MODELOS 3 e 4
4
CODMOD=89
CODMODH=∅
DIREÇÃO=∅
PASSO=∅3
CODME∅9
CODMH = ∅
DIREÇÃO=FD
MODELO : ∅2
MODELO : ∅1
MODELO : ∅∅
INCREM. MODELO
NUMOD:∅
INCREM. VOLTAS
REACENDE O TIRO DO CANHÃO
16
CODMODL=∅6
CODMODH=86
DELAY2=DELAY2−3
DELAY 2 : ∅3
DECREM. TIROS
TIROS : 04
VALMOD=VALMOD+∅2
VALMOD : '12'
DELAY1 = DELAY ½
DELAY 1 : 1
CODMODL = 84
CODMODH = 7
VOLTAS : '2∅'
VOLTAS = 0
AJUSTE DA ENERGIA
15
ENERGIA:∅
DELAY AUXILIAR

Exemplares Avulsos

Caro Amigo,

Se você não tem as revistas do número 9 até a 33 basta preencher o cupom ao lado (em letra de forma) e remetê-lo para nós, acompanhado de **cheque nominal ou Vale Postal em nome de Microdigital Eletrônica Ltda/ Divisão Microhobby à Caixa Postal 54096 - CEP 01136.**

Desejo receber os exemplares citados no cupom abaixo pelo preço de Cz$ 20,00.

Nome			
Endereço			
Bairro			
Cidade			
Estado		**CEP**	
Telefone			
Números Avulsos desejados:			
Cheque n.º	**Banco:**		**Vale Postal:** ☐

Prezado Leitor!
Responda a pesquisa ao lado, pois ela tem muito valor para nós.

Prezado Assinante:
Queremos saber o que você pensa de nós. Por isto, solicitamos que responda à pesquisa abaixo:

1) Qual a sua idade? ______________

2) Qual o seu nível de escolaridade?
1.º Grau ☐ 2.º Grau ☐
Superior Incompleto ☐ Superior Completo ☐
Outros ☐

3) Qual a sua profissão? ______________

4) Qual o seu equipamento?
Sinclair ☐ Spectrum ☐ Apple ☐

5) Onde você utiliza seu equipamento?
Casa ☐ Trabalho ☐ Escola ☐

6) Como conheceu Microhobby?
Amigos ☐ Bancas ☐ Lojas ☐ Outros: ________

7) Onde você mais lê Microhobby? ______________

8) Qual o assunto de maior interesse?
Programas ☐ Artigos Técnicos ☐
Informação Geral ☐ Reportagens ☐
Dicas ☐

Fórmula 1

Neste incrível jogo, desenvolvido para o TK-2000, você poderá testar toda a sua habilidade, pilotando um carro de cinco marchas, em seis diferentes circuitos e aprender um pouco mais sobre as capacidades gráficas deste micro.

Cesar de Afonseca e Silva Neto
Wilson José Tucci

Neste número iremos explorar um pouco mais as capacidades gráficas do TK-2000, através de um jogo que as utiliza através de traços e shapes. São mostrados seis modelos de circuitos (pistas) em que poderão ser realizadas as atividades. A descrição de cada um destes circuitos está contida nas linhas DATA do final do programa.

Escolhido o circuito, passaremos a controlar um carro (SHAPE), durante um tempo estipulado. No decorrer deste, teremos que desviar de obstáculos e realizar manobras a fim de dar o maior número possível de voltas no circuito. Terminado este intervalo, o bom motorista receberá um tempo extra como bonificação, e poderá continuar na corrida por mais algum tempo.

Recebida a bonificação, chega-se à hora de se fazer a avaliação do motorista. Nesta avaliação são levados em conta o número de voltas dadas e a dificuldade do circuito (circuitos com obstáculos rendem mais pontos). Logo a seguir, verifica-se o histórico de recordes (arquivo seqüencial) com o objetivo de analisar se algum foi superado. Caso tenha sido, o piloto poderá gravar as suas iniciais, juntamente com o seu total de pontos ("score").

Técnica empregada

Você perceberá que o programa é bastante rápido, embora não utilize nenhuma rotina especial, em Linguagem de Máquina.

Se utilizássemos métodos convencionais na comparação do carro com os limites dos circuitos, o programa se tornaria muito lento e inflexível. Lento, pois teria que realizar inúmeras comparações a fim de verificar se o carro colidiu com a pista, e inflexível, pois novos circuitos não poderiam ser acrescentados com tanta facilidade.

O artifício empregado reside na posição de memória 234. Esta posição é utilizada como um "contador de colisões", no que diz respeito a SHAPE-TABLES. Sendo assim, só precisamos realizar quatro comparações para saber se o carro bateu nos limites da pista ou não (linha 740).

Repare que existe mais do que um valor possível de ser encontrado na posição 234. Isto se deve aos diferentes valores de rotação (ROT).

Este número também está relacionado com o número de shapes que é utilizado. Portanto, torna-se impraticável controlá-lo com um maior número de shapes.

Introduzindo novos circuitos

Uma das grandes vantagens deste programa é a possibilidade de criar novos circuitos com facilidade.

A primeira coisa a ser feita é o desenho. Desenhe o circuito em uma folha de papel milimetrado e marque os pontos principais que serão armazenados em linhas DATA. Quanto maior for o número de pontos destinados ao circuito, melhor será a sua aparência no vídeo, principalmente nos casos em que existem curvas. Feito isto, já podemos iniciar a codificação do circuito.

Por exemplo, um circuito que apresente uma borda ao redor de toda a tela teria o seguinte início de codificação:

0,0,278,0,278,158,0,158,0,0,-2,... e assim por diante. Repare que o -2 tem a finalidade de indicar a rotina de desenho que diz que o próximo ponto não deverá ser ligado ao anterior. Sendo assim, sempre que o seu circuito apresentar pontos que devam ser descontínuos, estes deverão estar separados pelo valor -2.

Em seguida, utilize uma outra linha DATA a fim de armazenar as variáveis de controle do circuito, na seguinte ordem:

TL,X1,Y1,CD,LX,LY,V,NO,HO

A primeira variável desta lista ***(TL)*** contém o valor do tamanho da linha de partida. As variáveis ***X1*** e ***Y1*** deverão conter as coordenadas vertical e horizontal da posição inicial do carro no circuito, respectivamente. ***CD*** representa a dificuldade do circuito, devendo ser um número entre 1 e 8, já que ao escolhermos a opção de obstáculos, o valor 2 é acrescentado a CD, totalizando 10. ***LX*** e ***LY*** são as coordenadas do ponto mais alto da linha de partida do circuito. ***V*** contém o valor estimado para a distância mínima que deve ser percorrida pelo carro, a fim de que uma volta seja contada. Este valor evita que motoristas espertos cortem o caminho através das paredes. ***NO*** representa o número de obstáculos a serem acrescentados, no caso desses serem escolhidos. Finalmente, temos ***HO,*** altura de cada obstáculo, e as coordenadas de cada um dos ***NO*** obstáculos.

Entrando com o programa

Antes de mais nada, você deverá dizer ao TK-2000 que este programa será utilizado em outra área de memória, para que se evite conflitos com a página 1 de texto e gráficos. Isto é obtido com os comandos da listagem 1. Feito isto, você pode começar a digitar a listagem 2.

Estando o programa totalmente carregado, grave-o sob o nome de sua preferência e prossiga com a entrada da listagem 3, que corresponde aos dados do carro a ser utilizado pelo programa BASIC. Para tanto, entre em Linguagem de Máquina (LM) e digite os bytes da figura da seguinte maneira:

*330:01 00 04 00 27 37 ... 00 (RETURN)

Volte ao BASIC através de CTRL-C + RETURN e grave esta entrada sob o nome de CARRO, da seguinte maneira:

BSAVE CARRO,A$330,L$16

Antes de rodar o programa, você deverá digitar RUN 1350 e criar no seu disco o arquivo de recordes. Páre a execução e agora você já poderá rodar o programa e verificar os seis circuitos que estão disponíveis. Um deles está representado na figura 1.

Quando você estiver pilotando o carro, as teclas terão a seguinte função:

A e Z — Mudança de marchas.
← e → — Mudança da direção do carro.

Mais adiante existe uma tabela contendo as principais variáveis utilizadas neste programa, que visa possibilitar a você, uma melhor compreensão.

Tabela de Variáveis Utilizadas

AM	Análise do motorista (numérica).
AM$	Análise do motorista (alfanumérica).
BO	Quantidade de BÔNUS recolhido.
BT	Número total de batidas durante o percurso.
BX ( )	Matriz contendo as coordenadas X dos obstáculos do circuito escolhido.
BY ( )	Matriz contendo as coordenadas Y dos obstáculos do circuito escolhido.
C	Contém o valor 1 se o carro cruzou a linha de chegada, ou 2 se o carro bateu em alguma parede.
CD	Dificuldade do circuito escolhido.
DB	Contém o valor 2, sendo adicionada à dificuldade do circuito escolhido, no caso de haver obstáculos.
DV	Distância viajada durante o percurso.
LX	Coordenada X da linha de partida.
LY	Coordenada Y da linha de partida.
M	Marca em andamento.
NA$	Iniciais do nome do motorista que bate o recorde.
NA$ ( )	Matriz que contém as iniciais dos recordistas para cada circuito.
NO	Número de obstáculos no circuito.
NC	Número do circuito sendo mostrado.
PI	O valor 3,14159/32, utilizado nas equações de seno e cosseno.

▷

EXPLORANDO O TK 2000

R	Rotação do shape table (carro).
SC	"Score" do percurso.
SC ( )	Matriz contendo os recordes de cada circuito.
T	Contém o valor 4, representando o número de vezes que o programa tem que realizar o laço de decrementar TI (tempo).
TI	Tempo que ainda resta.
V	Contém a distância mínima que o motorista deve percorrer a fim de marcar uma volta.
VLT	Número de voltas dadas durante o percurso.
VX	Coordenada X antiga do carro.
VY	Coordenada Y antiga do carro.
X	Nova coordenada X do carro.
XS	Valor adicionado a X quando o carro muda de direção, utilizado nas equações de seno e cosseno.
Y	Nova coordenada Y do carro.
YS	Valor adicionado a Y quando carro muda de direção, utilizado nas equações de seno e cosseno.

LISTAGEM 1

```
10  REM  ** SETUP TK/2000 **
20  LOMEM: 16384: REM  ACIMA DA
     MA'
30  POKE 103,1: POKE 104,64: REM
     INICIO DO PROGRAMA '$2001'
40  POKE 16384,0: REM  NECESSARIO
```

LISTAGEM 2

```
100  REM

F O R M U L A - I

110  REM  CARREGAR SHAPE TABLE
120  PRINT  CHR$ (4)"BLOAD CARRO,
     A$330": GOSUB 1480:B = 0
130  REM  ROTINA 'INKEY$'
140  POKE 768,32: POKE 769,67: POKE
     770,240: POKE 771,133: POKE
     772,58: POKE 773,96
150  REM  TELA DE APRESENTACAO
160  TEXT : HOME
170 KEY = 768
180 M$ = " ### F O R M U L A - I
     ### - UM JOGO ONDE VOCE PODE
     RA' TESTAR TODAS AS SUAS HAB
     ILIDADES - PRESSIONE QUALQUE
     R TECLA PARA COMECAR  "
190  INVERSE
200  VTAB 1: PRINT  SPC( 40): FOR
     I = 2 TO 22: VTAB I: HTAB 1:
     PRINT " ";: HTAB 40: PRINT
     " ": NEXT : VTAB 23:PRINT  SPC(
     40)
210  VTAB 4: HTAB 10: PRINT  SPC(
     22)" ": HTAB 10: PRINT "  F
     O  R  M  U  L  A  ": HTAB 1
     0: PRINT  SPC( 23)
220  VTAB 9: HTAB 18: PRINT  SPC(
     5): PRINT : FOR I = 1 TO 4:HTAB
     19: PRINT "   ": NEXT : HTAB
     18: PRINT  SPC( 5)
230  VTAB 17: HTAB 10: PRINT  SPC(
     20): PRINT : HTAB 10: PRINT
     " ";: HTAB 29: PRINT " ": HTAB
     10: PRINT  SPC( 20): NORMAL
240  VTAB 18: HTAB 11: PRINT  LEFT$
     (M$,18):M$ =  MID$ (M$,2) +
     LEFT$ (M$,1): CALL KEY: IF
     PEEK (58) = 0 THEN  FOR I =
     1 TO 50: NEXT : GOTO 240
250  VTAB 21: HTAB 9: PRINT "VOCE
     QUER OBSTACULOS  (S/N) ?";:
     GET R$: IF R$ <  > "S" THEN
     BLOCK$ = "N"
260  REM  INICIALIZAR VARIAVEIS
270 DV = 700
280 M = 2:R = 32
290 H = 3.14159 / 32
300 VLT =  - 1:TIME = 300
310 X1 = 226:Y1 = 140:I = X1:J =
     Y1
320  POKE 232,48: POKE 233,3: REM
     INICIO DO SHAPE
330  REM  MOSTRAR OS CIRCUITOS
340  GOSUB 860
350 I = X1:J = Y1
360 X = 277:Y = 157
370 Z1 = 1
380  SCALE= 2
390  REM  INICIO DO JOGO
400  HOME : VTAB 21: HTAB 17: PRINT
     "PRONTO-": FOR W = 1 TO 1000
     : NEXT : VTAB 21:HTAB 17:PRINT
     "  VAI  ";""
410  VTAB 21: HTAB 32: PRINT "MAR
     CHA: ";M - 1
420  REM  DESENHAR CARRO - 1A.VEZ
430  XDRAW 1 AT X1,Y1: GOTO 700
440  REM  VERIFICAR SE O CARRO CR
     UZOU A LINHA DE CHEGADA OU B
     ATE NA PAREDE
450  IF X1 > LX - 20 AND X1 < LX +
     20 AND Y1 >  = LY AND Y1 < L
     Y + 42 THEN Z = 1: IF DV > V
     THEN VLT = VLT + 1: VTAB 21
     : PRINT "VOLTA: ";VLT:DV = 0

460  IF X1 < LX - 20 THEN CO = 0
470 U =  INT ( RND (1) * 2): IF U
     = 2 THEN U =  - 1
480  IF Z <  > 0 THEN 530
490  REM  BATEU O CARRO
500  ROT= 0:M = 2:BT = BT + 1
510  PRINT  CHR$ (7): PRINT  CHR$
     (7): REM  BATEU O CARRO
520 TI = TI - 4:T = 3: VTAB 21: HTAB
     32: PRINT "MARCHA: ";M - 1
530  ROT= R + 16
540 Z = 0
550  GOTO 570
560  REM  LER O TECLADO
570  CALL 768:P =  PEEK (58)
580  IF P = 136 THEN R = R - 4: IF
     R < 0 THEN R = 60
590  IF P = 149 THEN R = R + 4: IF
     R > 64 THEN R = 4
600  IF P = 193 AND M < 6 THEN  HTAB
     32: VTAB 21:M = M + 1: PRINT
     "MARCHA: ";M - 1
610  IF P = 218 AND M > 2 THEN  HTAB
     32: VTAB 21:M = M - 1: PRINT
     "MARCHA: ";M - 1
620 XS = (M * 2) * ( COS (R * H))
     :YS = (M * 2) * ( SIN (R * H
     ))
630 DV = DV + M
640 I = X1:J = Y1:X1 = I + XS: IF
     X1 >  = X THEN X1 = X
650  IF X1 < Z1 THEN X1 = Z1
660 Y1 = J + YS: IF Y1 > Y THEN Y
     1 = Y
670  IF Y1 < Z1 THEN Y1 = 0
680  IF TI < 0 THEN TI = 0:T = 3
690 T = T + 1: IF T = 4 THEN TIME
     = TIME - 1:T = 0: VTAB 21:HTAB
     16: PRINT "TEMPO: ";TI;"
     ": IF TIME =  < 0 THEN  HTAB
     22: VTAB 21: PRINT "0  ": GOTO
     760
```

```
700  XDRAW 1 AT I,J
710  ROT= R + 16
720  XDRAW 1 AT X1,Y1
730  REM  VERIFICAR COLISAO DE SH
     APES
740  IF  PEEK (234) (  ) 55 AND PEEK
     (234) (  ) 88 AND  PEEK (234
     ) (  ) 100 AND  PEEK (234) (
      ) 66 THEN 450
750  GOTO 570
760 B1 = B1 + 1: IF B1 = 2 THEN HOME
     : FOR EG = 1 TO 5: VTAB 21:HTAB
     15: PRINT "FIM DE JOGO": FOR
     I = 1 TO 400: VTAB 21: HTAB
     15: PRINT "             ": FOR
     I = 1 TO 240: NEXT : NEXT:GOTO
     980
770  IF VLT ( 3 THEN 760
780  IF VLT ) = 3 THEN  INVERSE
     : VTAB 23: HTAB 15: PRINT "B
     ONUS: ";
790 BO =  INT (V * VLT / 300) * 1
     0
800  IF BLOCK$ (  ) "N" THEN BO =
     BO + 30
810  REM  BONUS
820  FOR NB = 0 TO BO STEP 10: VTAB
     23: HTAB 22: PRINT NB: NEXT

830 TIME = BO:BT = BO
840  NORMAL
850  GOTO 700
860  REM  CONTROLE DE ESCOLHA DOS
      CIRCUITOS
870 WC = 1
880  IF BLOCK$ (  ) "N" THEN HC =
     HC + 2
890  GOSUB 1550
900  VTAB 21: HTAB 2: PRINT "APER
     TE (FIRE) P/ ESCOLHER UM CIR
     CUITO": VTAB 22: HTAB 8: PRINT
     "QUALQUER OUTRA P/ CONTINUAR
     "
910 NC = HC + CD: VTAB 23: HTAB 1
     : PRINT "CIRCUITO #";WC;" DI
     FICULDADE (1-10) = ";NC;"
     "
920 NC = 0
930  CALL KEY:CK =  PEEK (58): IF
     CK = 0 THEN 930
940  IF CK = 174 THEN  RETURN : REM
      FIRE
```

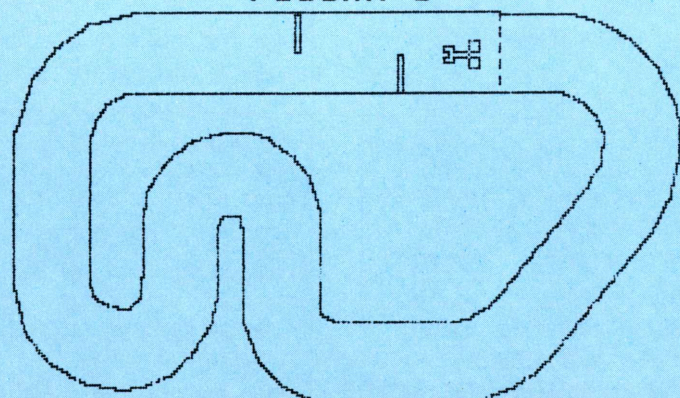
```
950 WC = WC + 1: IF WC = 7 THEN W
     C = 1: RESTORE
960  GOTO 890
970  HPLOT BX,BY TO BX,BY - HO TO
     BX + 2,BY - HO TO BX + 2,BY TO
     BX,BY: RETURN
980  REM  MOSTRAR ANALISE DO MOTO
     RISTA
990  TEXT
1000  INVERSE : HTAB 12: PRINT "P
     ERFORMANCE DO MOTORISTA":NORMAL

1010  PRINT : PRINT "---)CIRCUITO
     #";WC
1020  PRINT : PRINT "---)# DE VOL
     TAS: ";VLT
1030  PRINT : PRINT "---)TEMPO ME
     DIO POR VOLTA: ";: IF VLT =
     0 THEN  PRINT "300 SEG.": GOTO
     1050
1040  PRINT  INT ((300 + BT) / VL
     T)" SEG."
1050  PRINT : PRINT "---)# DE BAT
     IDAS: ";BT
1060  IF BLOCK$ (  ) "N" THEN CD =
     CD + 2
1070  PRINT : PRINT "---)NIVEL DE
      DIFICULDADE DO CIRCUITO :";
     CD
1080 AM = VLT + CD
1090  IF AM ( 8 THEN AM$ = "CHAME
     M UMA AMBULANCIA !": GOTO 11
     40
1100  IF AM ( 11 THEN AM$ = "CONT
     INUE TENTANDO": GOTO 1140
1110  IF AM ( 15 THEN AM$ = "VOCE
      DIRIJE BEM": GOTO 1140
1120  IF AM ( 19 THEN AM$ = "MUIT
     O BOM, MESMO !": GOTO 1140
1130 AM$ = "AYRTON SENNA QUE SE C
     UIDE !"
1140  VTAB 13: PRINT "---)AVALIAC
     AO :";AM$
1150 SC = V * VLT
1160  IF BLOCK$ (  ) "N" THEN SC =
     SC + (50 * VLT)
1170  PRINT : PRINT "---)RESULTAD
     O FINAL :";SC
1180  IF SC ) SC(WC) THEN  GOSUB
     1250: REM  NOVO RECORDE
1190  FOR WW = 1 TO 5000: NEXT:GOTO
     1410
```

```
1200  PRINT : HTAB 2: PRINT "ESCO
     LHA : (T)ERMINAR OU (RET) P/
      OUTRA"
1210  GET R$
1220  IF R$ = "T" THEN  TEXT : HOME
     : END
1230  CLEAR
1240  GOTO 160
1250  VTAB 17: HTAB 1: CALL  - 95
     8
1260  FOR I = 1 TO 10: VTAB 17: HTAB
     13:
1270  PRINT "NOVO RECORDE";: HTAB
     13: PRINT "            ": FOR
     WW = 1 TO 100: NEXT
1280  NEXT
1290  VTAB 19: HTAB 10: PRINT "EN
     TRE COM SUAS INICIAIS"
1300  VTAB 21: HTAB 18: PRINT  CHR$
     (95); CHR$ (95); CHR$ (95)
1310  FOR I = 0 TO 2
1320  VTAB 21: HTAB 18 + I: GET A
     $: PRINT A$:NA$ = NA$ + A$
1330  NEXT
1340 SC(WC) = SC:NA$(WC) = NA$
1350  PRINT  CHR$ (4)"OPEN FI.REC
     ORDE"
1360  PRINT  CHR$ (4)"WRITEFI.REC.
     ORDE"
1370  FOR I = 1 TO 6
1380  PRINT SC(I): PRINT NA$(I)
1390  NEXT
1400  PRINT  CHR$ (4)"CLOSE"
1410  HOME : HTAB 10: PRINT "HIST
     ORICO DOS RECORDES"
1420  VTAB 5: PRINT "CIRCUITO # "
     ;: HTAB 17: PRINT "SCORE";:HTAB
     27: PRINT "DETENTOR": PRINT

1430  FOR I = 1 TO 6
1440  PRINT "CIRCUITO # ";I;: HTAB
     17: PRINT SC(I);: HTAB 32:PRINT
     NA$(I)
1450  PRINT
1460  NEXT
1470  PRINT : GOTO 1200
1480  PRINT  CHR$ (4)"OPEN FI.REC
     ORDE"
1490  PRINT  CHR$ (4)"READ FI.REC
     ORDE"
1500  FOR I = 1 TO 6
```

EXPLORANDO O TK 2000

```
1510  INPUT SC(I): INPUT NA$(I)
1520  NEXT
1530  PRINT  CHR$ (4)"CLOSE"
1540  RETURN
1550  HGR : HCOLOR= 3
1560  READ X
1570  READ Y
1580 XV = X:YV = Y
1590  HPLOT X,Y
1600  READ X
1610  IF X =  - 2 THEN 1560
1620  IF X <  >  - 3 THEN 1670
1625  READ TL
1630  READ X
1640  READ Y:YV = Y
1650  HPLOT X,Y TO X,Y + 3:Y = Y +
     6: IF Y > YV + TL THEN  READ
     X1,Y1,CD,LX,LY0,V,NO,HO: FOR
     I = 1 TO NO: READ BX(I),BY(I
     ): NEXT : GOTO 1710
1660  GOTO 1650
1670  READ Y
1680  HPLOT XV,YV TO X,Y
1690 XV = X:YV = Y
1700  GOTO 1600
1710  IF BLOCK$ = "N" THEN  RETURN

1720  FOR I = 1 TO NO: HPLOT BX(I
     ),BY(I) TO BX(I),BY(I) - HO TO
     BX(I) + 2,BY(I) - HO TO BX(I
     ) + 2,BY(I) TO BX(I),BY(I):NEXT
     : RETURN
2000  REM  DADOS DOS CIRCUITOS
2010  DATA    0,0,278,0,278,158,0,
     158,0,0,-2,84,0,84,84,180,84
     ,-2,42,42,42,122,222,122,222
     ,42,126,42,-3,40,222,124
```

```
2020  DATA  222,140,4,222,126,395

2030  DATA  4,14,180,158,100,158,
     60,138,140,138
2040  DATA    0,0,278,0,278,158,0
     ,158,0,0,-2,42,42,42,118,134
     ,118,-2,88,0,88,84,-2,182,15
     8,182.90,-2,134,118,134,42,2
     30,42,230,118,-3,40,230,0
2050  DATA    222,14,7,222,0,300
2060  DATA  3,14,50,132,132,132,9
     0,158
2070  DATA    0,48,48,0,230,0,278
     ,48,278,110,230,158,48,158,0
     ,110,0,48,-2,48,66,66.42,212
     ,42,230,66,230,92,212,116,66
     ,116,48,92,48,66,-3,40,212,0

2080  DATA    214,20,2,212,0,150
2090  DATA   4,14,80,130,120,158,
     160,130,200,158
2100  DATA   0,0,278,0,278,159,0,
     159,0,0,-2,42,126,42,42,222,
     42,-2,84,159,84,84,90,84,90,
     159,-2,278,84,180,84,-2,222,
     126,132,126,132,42,-3,40,222
     ,3
2110  DATA    226,20,6,222,0,400
2120  DATA   5,14,180,14,100,14,1
     40,42,180,98,222,124
2130  DATA   0,0,278,0,278,159,0,
     159,0,0,-2,0,84,90,84,-2,42,
     126,228,126,228,42,-2,180,0,
     180,84,-2,42,42,132,42,132,1
     26,-3,40,228,126
2140  DATA    226,140,5,227,126,3
     60
```

```
2150  DATA  4,14,42,42,82,14,88,9
     8,42,126
2160  REM  DADOS DA SUPER PISTA
2170  DATA    200,10,60,10,39,15,
     26,24,17,35,10,55,10,111,15,
     130,24,141,38,149,50,151,62,
     149,72,145,80,138,88,126,90,
     117
2180  DATA  90,107,90,91,92,86,96
     ,85,99,87,100,91,100,125,106
     ,140
2190  DATA  115,150,126,155,136,1
     56,180,156,195,154,206,150,2
     14,145,220,140,260,92,267,79

2200  DATA  270,66,270,55,268,44,
     262,32,252,21,240,14,230,11,
     200,10
2210  DATA  -2
2220  DATA  200,40,61,40,53,41,46
     ,45,42,51,40,57,40,110,42,11
     7,47,120,51,121
2230  DATA  56,120,60,116,61,108,
     61,89,63,76,71,65,80,58,91,5
     5,101,55,110,58
2240  DATA  120,65,126,72,130,84,
     130,118,131,124,135,126,181,
     126,189,124,196,120,203,113

2250  DATA  235,75,240,67,241,60,
     240,54,237,48,231,43,225,40,
     200,40
2260  DATA   -3,30,200,10,200,25,
     8,200,10,300
2270  DATA   2,10,160,40,120,20
```

LISTAGEM 3

```
*330.345

0330- 01 00 04 00 27 37 2E 36
0338- 3E 37 2E 25 2C 2D 36 3F
0340- 24 24 2C 24 37 00
```

Uma introdução, através do TK-90X

Além de desenvolver a criatividade da criança, o LOGO aumenta o poder de concentração e agiliza o raciocínio.

Elaine Kelly Lima

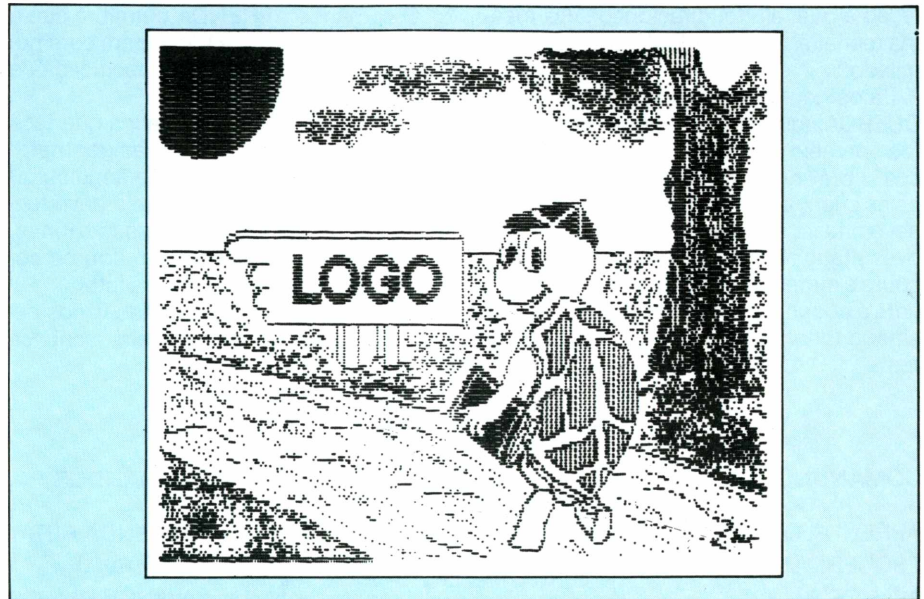
Ao final da década de 60 um grupo l - derado pelo matemático Seygmour Papert desenvolveu, no Instituto de Tecnologia de Massachusetts, a primeira versão do LOGO. O objetivo deles era criar uma linguagem que pudesse ser usada no ensino de crianças e para isso basearam-se em estudos realizados por Jean Piaget que comprovavam não ser aconselhável ensinar crianças a programar em determinada linguagem, pois elas sentem dificuldades para compreender conceitos abstratos.

Segundo o grupo, a linguagem deveria ajudar a criança a desvendar o mundo sem exigir que ela decorasse comandos. Pelo contrário, o aprendizado deveria ser espontâneo e à medida que houvesse necessidade de ampliar seus conhecimentos, a própria criança se interessaria por descobrir as novas instruções e recursos oferecidos, com os quais seria capaz de inovar e melhorar as rotinas que quisesse desenvolver.

Como resultado final da pesquisa do grupo nasceu o LOGO, que conseguiu o sucesso imediato devido às suas características. E, atualmente é inúmera a quantidade de versões que a linguagem possui para diferentes equipamentos.

O TK-90X e o LOGO

Por ser um micro acessível e oferecer recursos tais como cor e som, os micros da linha Spectrum como o TK-90X sempre tiveram um número muito grande de usuários.

Entre as usuários que adquirem o TK-90X muitas preferem deixar os filhos afastados do equipamento, ou seja, o menor tempo possível em contato com ele, por temerem que o televisor cause tensão ocular à criança. Porém, os pais não devem se preocupar com isso, pois raramente a criança consegue ficar muito tempo concentrada em alguma coisa e desta maneira, elas não estão sujeitas a este tipo de problema.

A única coisa pela qual os pais devem ficar alertas é que algumas linhas de televisores a cores fabricados antes de 1970 emitem uma pequena carga de radiação, que é prejudicial. Por este motivo é bom certificar-se de que o seu televisor a cores utilizado tenha sido fabricado posteriormente à esta data.

Agora, com o desenvolvimento de uma versão do LOGO para esse equipamento, ficou provado que no processo de educação infantil este pode ser muito favorável às crianças, pois mistura seus recursos com tecnologia desenvolvida especialmente para o ensino. Além disso introduz a criança no universo da computação de maneira gostosa e descontraída, o que a entusiasma, criando um novo interesse quanto à utilização do micro.

Utilizando o LOGO

Tendo como objetivo primordial ambientar a criança com processamento de dados sem forçar sua cabecinha, os comandos da linguagem foram nomeados de acordo com o conceito de que a criança não seja obrigada a decorá-los.

E dessa forma, os comandos foram transformados em palavras do seu vocabulário habitual, que geram respostas cujo sentido tenha a mesma idéia do significado da palavra.

Para que a criança tenha algo visível à sua frente que elhe forneça a idéia de execução das suas "ordens" (comandos), o LOGO possui uma tartaruga que á a responsável pelo traçamento dos desenhos.

O método de aprendizagem da linguagem é simples, sendo que a criança assimila os comandos através da prática própria, ou seja, ela testa várias vezes os comandos e instruções e verifica, à cada nova tentativa, a resposta dada pela tartaruga.

É bom lembrar que desta maneira a assimilação dos conhecimentos será muito mais rápida e duradoura, visto que a criança está aprendendo por si mesma.

De acordo com o dito acima ao iniciar o contato da criança com o micro você deve mostrar a ela um comando, deixando-a testá-lo. Desta maneira, ela irá errar várias vezes, e assim se ambientará com os limites da "tartaruga".

Alguns dos principais comandos da versão LOGO para o TK-90X são:

Comando	Utilidade
VENHA	Faz com que a tartaruga apareça na tela
LIMPE TELA	Apaga tela
AVANCE N	Faz a tartaruga avançar N passos
VOLTE N	Faz a tartaruga voltar N passos
DIREITA G	Faz a tartaruga girar G graus à direita
ESQUERDA G	Faz a tartaruga girar G graus à esquerda

Por exemplo, no início a primeira instrução que será apresentada à criança será VENHA e desta maneira, ela perceberá que este comando provocará o aparecimento de uma figura no centro da tela, figura que não é nada mais nada menos, do que a famosa tartaruga que acompanhará todas os traçados de cada desenho.

Vendo a tartaruga, a criança sentirá necessidade de aprender como fazê-la andar e assim, sentirá a necessidade de usar e desvendar os segredos do comando AVANCE que a faz avançar *n* passos na di-

reção à qual ela foi direcionada. Da mesma maneira, o comando VOLTE será assimilado.

Já os comandos como DIREITA e ESQUERDA, (que são utilizados pelos adultos por meio de conhecimentos geométricos sobre ângulos), devem ser aprendidos pelas crianças por meio de testes que elas devem fazer até que consigam identificar o resultado obtido. Uma vez tendo a noção sobre a dimensão do giro, a criança estará apta a coonhecer novos comandos ou ensinar a tartaruga a construir coisas maiores.

O comando APRENDA permitirá que a criança implemente a linguagem com novas estruturas que serão especificadas e nomeadas por elas.

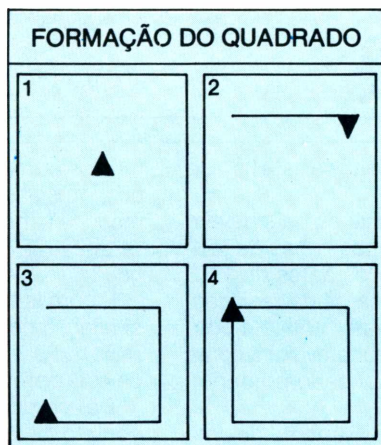
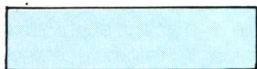
De início proponha à criança que faça rotinas para desenhar formas geométricas, tais como quadrados, retângulos ou triângulos e desta maneira, ela deve tentar, sózinha, desenvolver essas rotinas, pois se ela realmente assimilou os comandos tem capacidade para isto.

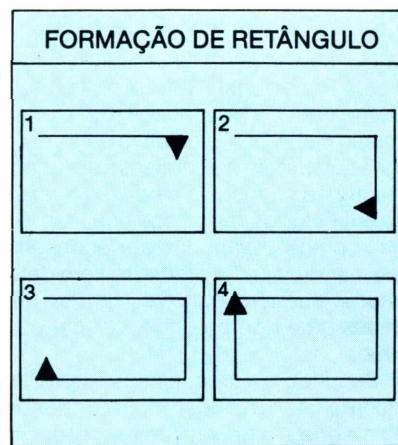
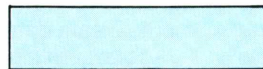
Depois de ter implementado vários pequenos desenhos à linguagem, a criança começa a juntá-los e assim idealiza projetos maiores e passa a perceber que o LOGO tem uma amplitude muito maior do que ela imaginava.

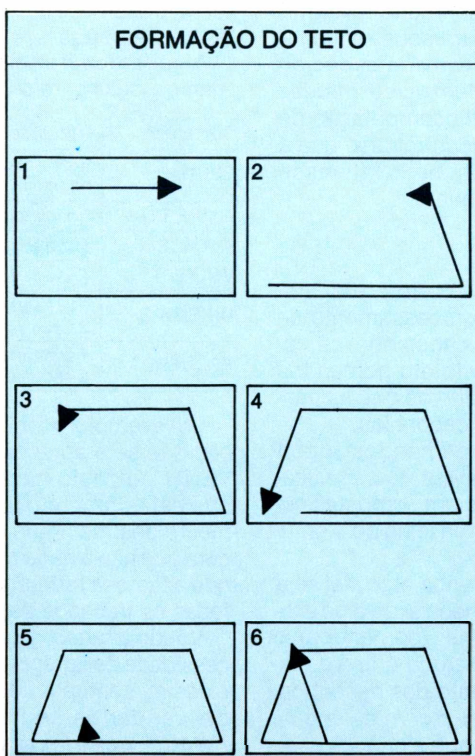
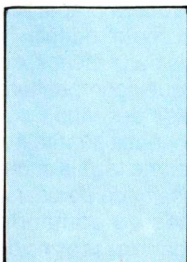
Rotinas para formar uma casa no vídeo

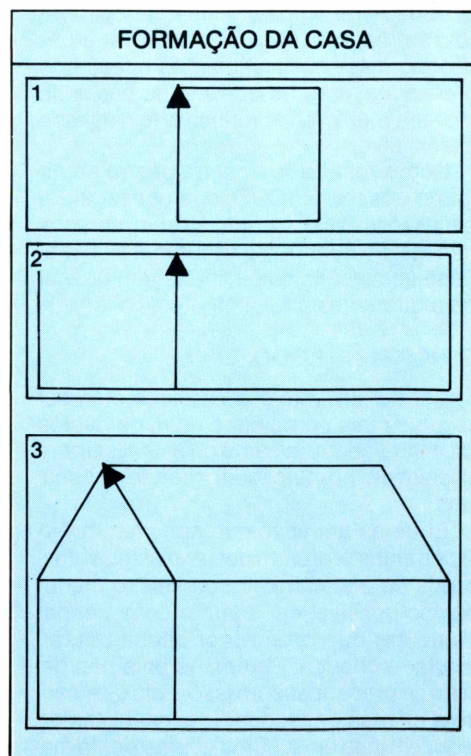
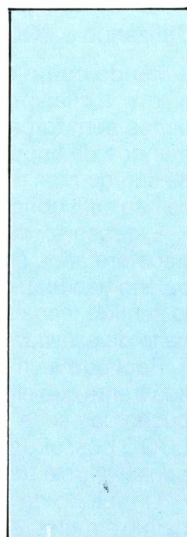
Através da junção de três rotininhas que são criadas, separadamente, a criança pode fazer o formato de uma casa na tela. Essas rotinas foram desenvolvidas somente com a utilização dos comandos básicos da linguagem.

COMANDOS

APRENDA QUADRADO
ESQUERDA 90
AVANCE 30
ESQUERDA 90
AVANCE 30
ESQUERDA 90
AVANCE 30
ESQUERDA 90
AVANCE 30
FIM

COMANDOS

APRENDA RETÂNGULO
DIREITA 90
AVANCE 50
DIREITA 90
AVANCE 30
DIREITA 90
AVANCE 50
DIREITA 90
AVANCE 30
FIM

COMANDOS

APRENDA TETO
DIREITA 90
AVANCE 50
ESQUERDA 135
AVANCE 30
ESQUERDA 45
AVANCE 45
ESQUERDA 60
AVANCE 25
ESQUERDA 120
AVANCE 30
ESQUERDA 127.5
AVANCE 27
FIM

COMANDOS

APRENDA CASA
QUADRADO
RETÂNGULO
TETO
FIM

Páre e verifique os endereços!

Christiano A. C. Nasser

Várias vezes, quando estamos programando em Linguagem de Máquina necessitamos parar e verificar o conteúdo do acumulador e dos registradores, para em seguida continuarmos o processamento e em muitas ocasiões não dispomos de um programa mapeador que apresente todos os endereços usados e o resultado das operações aritméticas.

Para estes casos, a rotina abaixo pode ser muito útil:

```
PARE
     BRK
     NOP
     PLA
     PLA
     JSR $FF3F
     RTS
```

A qualquer momento, desejando parar o programa, mande-o para a JSR pare. O processador vai reconhecer o BRK e em seguida, mostrar o acumulador, os registradores X e Y, e algumas FLAGS.

Como funciona

O que acontece no processador quando um BRK é encontrado, é que o TK-2000, ignora o próximo byte (no caso NOP) e coloca o endereço do primeiro PLA nas posições $3A E$3B, guardando as flags e STACKs nas posições de $45 até $49.

Continuando o processamento, o TK-2000 pega o conteúdo das posições $3A e $3B, monta o endereço e continua a rodar o programa a partir daí.

Os PLAs são necessários para recuperar os dois bytes que são usados no STACK. (Posiciona a próxima linha a ser executada no programa).

Os próximos dois bytes serão usados pela instrução seguinte, RTS retornando para seu programa.

A subrotina $FF3F é usada para restabelecer os conteúdos dos registradores, do acumulador e status.

Vamos verificar internamente as intrincadas "malhas" do complexo da rotina $FF3F.

```
FF3F: AD F3 07    LDA  STATUS
FF42: 48          PHA
FF43: A5 45       LDA  ACC
FF45: 4C 3A FC    JMP  RSTR1
FF48: EA          NOP
FF49: EA          NOP
FF4A: 4C 9D FB    JMP  SAVE1
FF4D: 8C F2 07    STY  YREG
FF50: 08          PHP
FF51: 68          PLA
FF52: 8D F3 07    STA  STATUS
FF55: BA          TSX
FF56: 8E F4 07    STX  STNP
FF59: D8          CLD
FF5A: 60          RTS
```

No caso:

```
STATUS — $07F3
ACC    — $45
RSTR1  — $FC3A
SAVE1  — $FB9D
YREG   — $07F2
STNP   — $07F4
XREG   — $07F1
```

Nas três primeiras linhas, armazenamos o STATUS no STACK. Em seguida, temos JMP RSTR1, que é a segunda rotina utilizada. Vamos conferir:

```
FC3A: AE F1 07    LDX  XREG
FC3D: AC F2 07    LDY  YREG
FC40: 28          PLP
FC41: 60          RTS
```

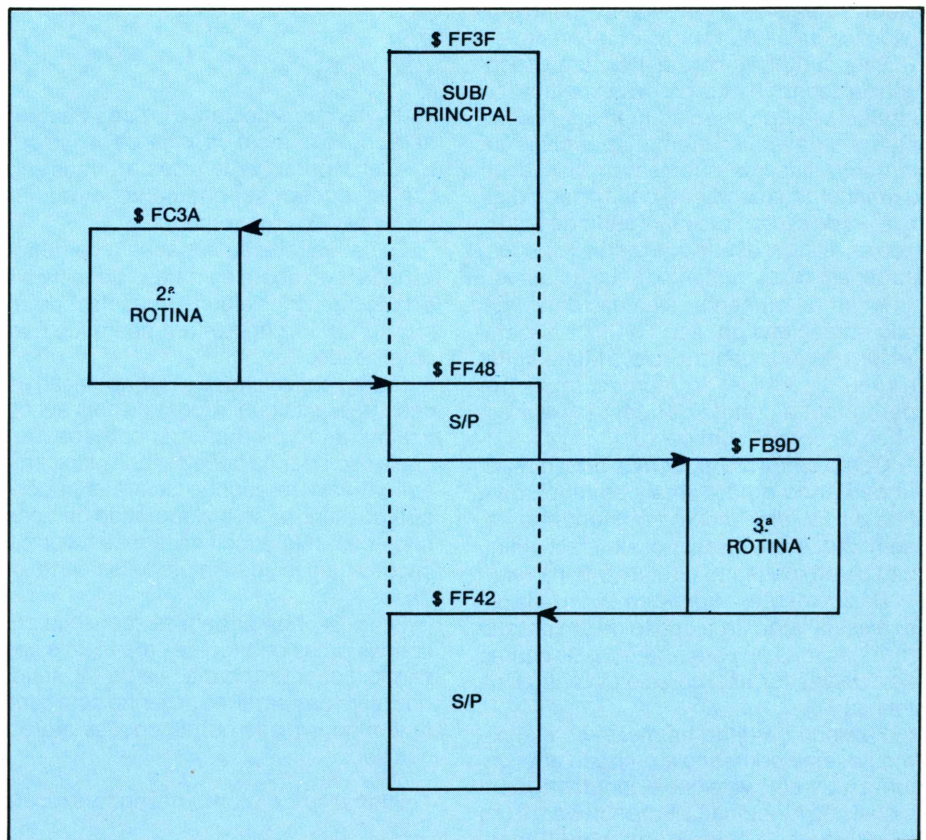
Nesta rotina, restauramos os valores de X, Y e do STATUS no STACK. Em seguida, voltamos à rotina principal.

Em $FF4A, vamos para mais uma rotina, onde salvaremos o acumulador e o registrador X.

```
FB9D: 85 45       STA  ACC
FB9F: 8E F1 07    STX  XREG
FBA2: A5 45       LDA  ACC
FBA4: 8D F0 07    STA  AREG
FBA7: 4C 4D FF    JMP  SAV3
```

Esta última linha, mada de volta à rotina principal, fechando a "malha". As linhas seguintes salvam o valor de T em YREG; o registrador de STATUS em "STATUS" ($07F3), e o STACK POINTER em SPNT.

Como podemos perceber, o que existe é um intricado sistema de subrotinas que se encaixam, para realizar uma só operação. Examine a figura 1, para entender melhor as rotinas que acabamos de ver.

FIGURA 1

Bibliografia

TK-2000/II - Entendendo a ROM - Geraldo Coen - Micromega - páginas 146, 154 e 155.

Light-Pen

A Microdigital colocou no mercado mais um periférico para tornar o seu TK-90X, cada vez mais, um micro voltado às suas necessidades.

Elaine Kelly Lima

Quando comparamos os micros aos sistemas maiores logo encontramos as restrições por eles apresentadas, tais como: a pequena capacidade de memória e a baixa-resolução gráfica, que provocam a redução do número de cores geradas. Além disso, o seu uso em aplicações que exigem a transmissão de informações via imagens, apresenta problemas relativos à introdução, manipulação e saída de dados no equipamento.

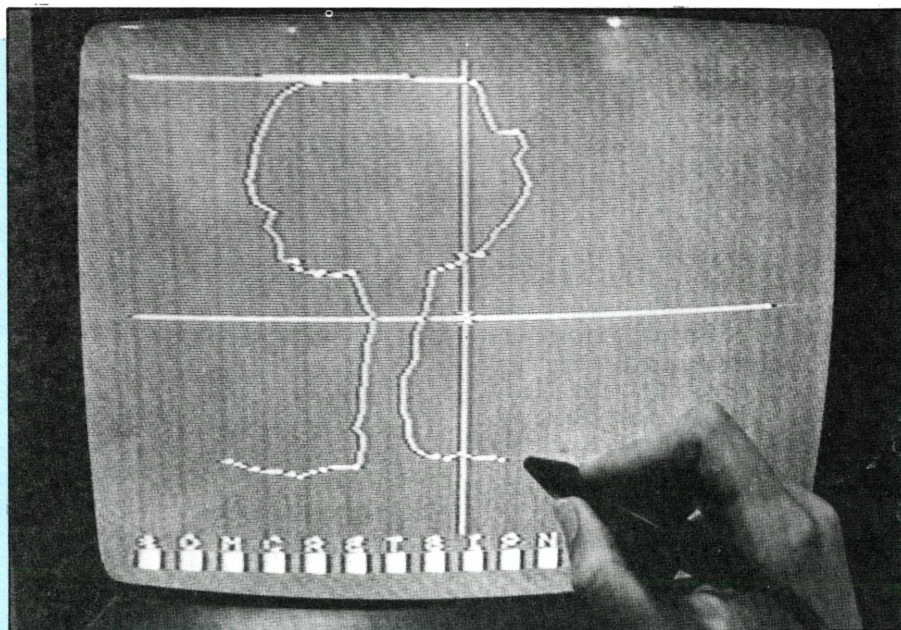
Para contornar isso e agilizar a execução de determinados processamentos, os fabricantes de sistemas gráficos pesquisaram e desenvolveram dispositivos de I/O (entrada/saída) que visavam auxiliar os usuários na elaboração de imagens digitais. Pôde-se comprovar, a partir de então, que os micros, desde que bem equipados, poderiam fazer muito por seus usuários.

A fim de aumentar as áreas de aplicação abrangidas por seu TK-90X e satisfazer os usuários deste micro, a Microdigital tornou disponível ao mercado mais um periférico: a caneta óptica, também conhecida como "light-pen".

Como definição, podemos dizer que este periférico obtém sinais emitidos pela tela da TV, enviando-os posteriormente, ao micro. Além disso, possibilita a definição de um desenho diretamente na tela.

Basicamente, a ***light-pen*** é utilizada para a escolha de uma opção de um programa de menus ou para a geração de figuras de diversas formas, cores e tamanhos, na tela da TV.

Fazendo parte do hardware do equipamento, este periférico é utilizado em conjunto a um software que o coordena e sem o qual não funcionaria. Este software é um programa que é adquirido, simultaneamente, com o periférico.

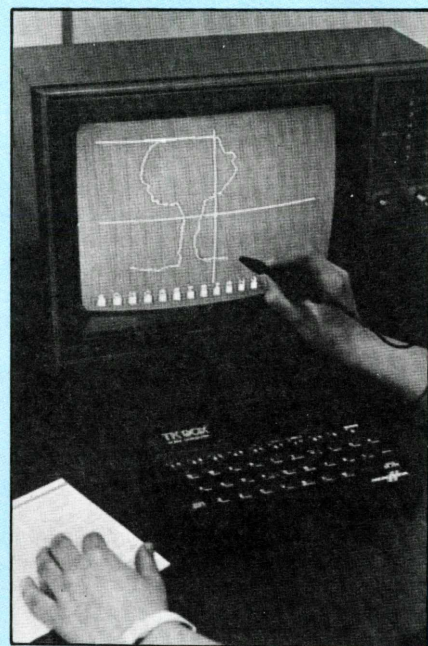
Quando a light-pen é utilizada para selecionar opções em um programa de menus deve-se ter o software carregado, na memória do equipamento. Em seu programa deverá constar a chamada para a rotina do software, que detectará a luz da TV e retornará com o número da linha, da opção que a caneta apontou. Além desta, outras subrotinas podem ser chamadas durante a execução de um programa.

Israel Teixeira

Uma mão, uma caneta e uma tela, a mágica do light-pen começa a surgir.

O manual que acompanha o periférico explica tudo sobre as rotinas do software e dá aos usuários alguns "macetes" de como tornar a ***light-pen*** um periférico bem empregado.

É bom observar que a ***light-pen*** não necessita de qualquer programa para ser utilizada, muito pelo contrário, o software que a acompanha pode ser utilizado como aplicativo, oferecendo diversas opções que possibilitarão a elaboração de desenhos com muita facilidade e sem complicações para leigos em processamento de dados.

Os desenhos podem ser conseguidos apenas pressionando-se uma tecla e passando a caneta sobre a tela da TV, sendo que além de permitir o desenho com caneta, à mão livre, as outras opções oferecidas são:

- eliminação do último desenho executado;
- traçamento de linha da origem ao destino, sendo que ambos são especificados pelo usuário;
- movimentação do posicionador de exibição;

Israel Teixeira

Os segredos da máquina são desvendados com a ajuda do pequeno 90.

- traçagem de círculo;
- traçagem de quadrilátero;
- preencher uma forma fechada com determinada cor;
- definição da cor da borda da tela;
- definição da cor do traçado;
- definição da cor do fundo da tela;
- apagar a tela;
- salvar e carregar telas em memória auxiliar;
- armazenar na memória até quatro páginas de vídeo;
- reapresentação ou movimentação de página em memória;
- traçagem de arco sendo definido, pelo usuário, três pontos;
- e inserção de texto à imagem.

Este periférico é acessível aos TK-90X de 16 e 48 Kbytes de RAM tendo larga utilização pessoal e profissional. Seu manual e sua utilização são fáceis e práticos. Portanto não se intimide e acople, mais este recurso, ao seu equipamento. A ***light-pen*** está sendo comercializada juntamente com a nova versão do TK-90X; o ***TK-90X Color-LP.***

O pacote Light-pen: software, caneta óptica e a interface.

CAPA

As linguagens de programação...

... responsáveis pelo intercâmbio de informações entre o homem e a máquina.

Elaine Kelly Lima
Marcos Lorenzi
Paulo Panagiotis Alvanos

O computador é um objeto extremamente "burro", é incapaz de resolver um problema sem que previamente alguém introduza, em memória, comandos específicos para este fim. Por este motivo, o intercâmbio de informações entre o Homem e a máquina vem se aprimorando desde o começo da computação até hoje.

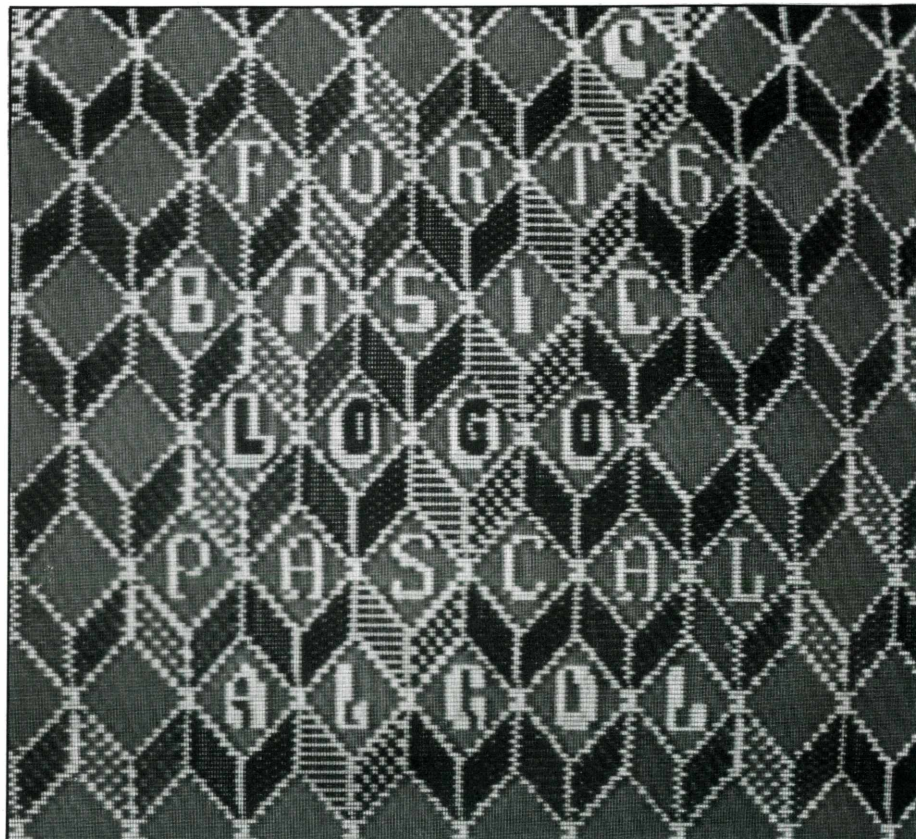
Levando em conta que o dialeto entendido pelo computador é extremamente diferente do usado por nós, podemos dizer que existem duas maneiras para a comunicação: aprende-se a Linguagem de Máquina ou usa-se um tradutor.

A programação em códigos de máquina se resume na introdução de números binários em determinados locais da memória do equipamento. Por este motivo, para programar a este nível, é necessário conhecer muito bem o hardware utilizado. Em código de máquina, os programas se tornam difíceis de serem compreendidos, assim como para se executar correções.

Usando um tradutor, a tarefa de programação fica muito mais simples e agradável. As linguagens traduzidas são classificadas em dois tipos: baixo-nível e alto-nível.

A primeira assemelha-se muito aos códigos de máquina e é difícil de se aprender, sendo que seu melhor exemplo é o Assembler, que utiliza mnemônicos seguidos de operandos. Esta linguagem muda completamente em se tratando de CPUs diferentes. O tradutor utilizado por ela é denominado Montador e traduz o programa-fonte (original) para um programa-objeto (códigos binários relativos às instruções do programa-fonte).

As linguagens de alto-nível são facilmente assimiladas por se assemelharem à humana. Seus tradutores podem ser: interpretadores ou compiladores.

A linguagem interpretada mais conhecida é o BASIC, cujo interpretador muitas vezes, mas não sempre, se encontra embutido na própria ROM do equipamento. Este tipo de tradutor leva o usuário a uma espécie de editor interno que permite fazer e alterar o programa com facilidade. Quando rodado, o programa é traduzido linha por linha, sendo que é impossível rodá-lo quando não se tem o interpretador na memória.

Na maioria das linguagens o tradutor é o compilador. A compilação é a obtenção de um programa-objeto proveniente da tradução do programa-fonte, sendo que o programa-objeto somente é criado quando nenhuma das regras da linguagem é quebrada. Caso isso ocorra, o compilador apresenta os erros, exibindo-os no vídeo ou na impressora. Normalmente, a sintaxe é conferida com base em uma biblioteca que complementa o compilador.

Cada linguagem possui um compilador específico a ela, e mesmo as que normalmente são interpretadas possuem compiladores. Devemos chamar o programa-objeto para podermos rodar o programa, já compilado, pois dessa forma, ele não necessitará de nenhum software para ser processado. Além de ser auto-suficiente, os programas compilados realizam seus processamentos com maior rapidez que os interpretados.

Para alteração de programas-objetos usa-se "debugs" ou ainda pode-se refazer o programa-fonte e compilá-lo novamente. Porém, isto causa, em determinados casos, muitos problemas.

Muitos compiladores necessitam de um "link-editor" para complementar sua tarefa. O "link-editor" irá juntar, em um único programa, as várias partes em que ele foi dividido durante a compilação.

Com essa pequena introdução das linguagens de programação, você poderá entrar no universo dos dialetos, onde estima-se existirem aproximadamente quatrocentas linguagens voltadas para diversas áreas.

Para optar por uma delas, é fundamental fazer uma comparação (das suas vantagens e desvantagens), assim como levar em conta as suas diferentes aplicações.

Desta forma, apresentaremos a seguir as principais características das linguagens mais conhecidas e um resumo sobre suas histórias.

Fortran (Fórmula Translation)

O Fortran, desenvolvido por um grupo da IBM em 1955, liderado por Jim Backus, é uma das primeiras linguagens de alto-

nível, que visa o processamento de fórmulas com uma certa facilidade.

Pelo fato de ser uma linguagem compilada, tem como característica principal seu processamento rápido.

Pode-se observar a sua utilização em grandes indústrias que usam um sistema para desenho de peças e posterior fabricação. Este sistema, conhecido por CAD-CAM, foi desenvolvido em Fortran, devido à necessidade de se trabalhar com um número muito grande de fórmulas.

A utilização de periféricos e seu manuseio é bem simples quando se trabalha em Fortran. Para a movimentação de dados e uso de variáveis, existem comandos e algumas regras a serem seguidas, por exemplo: na movimentação de dados para o meio exterior é necessário usar declarações de entrada e saída (I/O), tais como as instruções WRITE, READ e FORMAT. Já para especificação de variáveis, é necessário discriminar o seu tipo, que poderão ser reais ou inteiras. O computador entende como variáveis inteiras todas aquelas iniciadas pelas letras (I, J, K, L, M, N), caso contrário serão reais.

Exemplos de variáveis:
INTEI ... Variável INTEIRA
MICRO ... Variável Inteira
HOBBY ... Variável Real

Qualquer variável em Fortran deve ter, no máximo, cinco caracteres. Se por acaso ultrapassar este limite, ocorrerá um erro na hora da compilação do mesmo.

A tradução do programa é semelhante ao COBOL. É necessário editar, compilar e "LINKar" o programa antes de sua execução.

Linguagem BASIC

Criada em 1964 por John G. Kemeny e Thomas E. Kurtz (professores da Dartmouth College em Hanover, New Hampshire nos EUA), a linguagem BASIC foi inspirada na linguagem Fortran, para facilitar o aprendizado da programação de computadores e a execução de cálculos matemáticos. Atualmente ela é considerada a mais popular e utilizada em todo o mundo.

Originalmente o BASIC foi disponível para minicomputadores e posteriormente, seu uso chegou aos microcomputadores, onde atualmente estão seus principais usuários. Praticamente todos os minis e micros dispõem da linguagem BASIC.

Embora seja uma linguagem potente, o BASIC é fácil de se aprender e o objetivo de sua criação, para auxiliar o ensino do processamento de dados, foi alcançado. Com poucas horas de introdução qualquer pessoa consegue desenvolver algum programa simples.

O BASIC é uma linguagem interativa e de alto-nível, sendo que a maior parte de suas versões são interpretadas, ao invés de serem compiladas, apesar de existirem compiladores. Isto ocorre devido à facilidade oferecida pelos interpretadores.

O BASIC não tem uma versão padrão universalmente estabelecida, cada fabricante tem desenvolvido sua própria versão de acordo com o hardware dos equipamentos. Com isto torna-se muito difícil o conhecimento de todas as suas versões, das quais as mais conhecidas e utilizadas são:

— ***Sinclair;***
— ***Spectrum;***
— ***Apple;***
— ***TRS-80;***
— ***Microsoft.***

Pode-se dizer então que: o BASIC do Sinclair e do Spectrum são voltados aos micros pessoais; o do Apple possui inigualáveis recursos gráficos e do TRS-80 e o da Microsoft são voltados para micros profissionais.

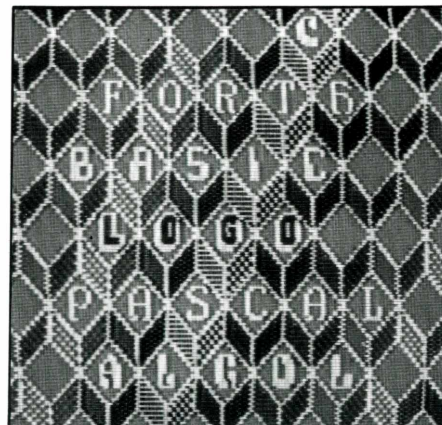
O BASIC conseguiu alcançar seu principal objetivo e hoje é a mais utilizada linguagem, tanto para o divertimento, como para aplicações comerciais. Pode-se dizer também que cada versão possui uma inovação, ou seja, algo de novo que expande a sua aplicação para as mais distintas áreas.

COBOL (Common Business Oriented Language)

O COBOL, linguagem de alto-nível criada em 1960, visava desde o seu início a área comercial, devido as suas inúmeras vantagens e fácil manuseio. O COBOL é uma das linguagens mais utilizadas em todo o mundo. Segundo uma pesquisa realizada nos Estados Unidos, as vagas (emprego) oferecidas para programadores COBOL superam-se em três vezes mais comparando-as com as vagas para todas as outras linguagens juntas. Com isto podemos notar em que posição o COBOL está dentre as linguagens mais procuradas.

Por que COBOL?..., esta é uma das perguntas que mais são feitas na hora de se implantar um sistema de computação; na hora de se fazer um programa comercial, ou na hora de se estudar uma determinada linguagem. Pensando em lhes dar uma resposta convincente, apresentaremos algumas das vantagens e desvantagens desta linguagem.

Desvantagens:

- não existe um padrão definido, dentre todas as versões do Compilador COBOL;

ou seja, um programa feito para uma família de computadores não funciona em uma família diferente, tendo que serem feitas alterações;
- o programa em COBOL ocupa muito espaço de memória, devido a sua auto-documentação.

Vantagens:

- fácil manutenção: aumentar, diminuir ou melhorar o programa;
- fácil controle de periféricos e de grandes quantidades de dados;
- permite a formatação de relatórios e telas, com uma relativa facilidade;
- o nome dos campos, dos registros e dos arquivos, podem ter a extensão máxima de 30 caracteres, facilitando, com isso, o entendimento do programa e sua documentação;
- rapidez no processamento;

Citadas algumas de suas vantagens e desvantagens, podemos ter, agora, uma noção do que uma linguagem como o COBOL pode nos fornecer.

Alguns exemplos de sua aplicação podem ser notados em: folhas de pagamento, controles de estoque, contabilidade, etc, campos em que o COBOL pode especialmente ser utilizado, pois necessitam de uma documentação mais rígida e a manipulação de muitos dados.

Trabalhando com o COBOL

a) Divisões de sua programação

O COBOL é dividido em quatro subdivisões principais, que são:

1 - IDENTIFICATION DIVISION: esta é a divisão de identificação. É através dela que podemos saber quem escreveu o programa; o local de seu desenvolvimento; a data da compilação; a data em que foi escrito o programa e os comentários gerais;

2 - ENVIRONMENT DIVISION: divisão de ambiente, ou seja, é nesta divisão que

são especificados os periféricos que serão utilizados e suas características; o equipamento em que o programa foi compilado e em qual foi executado;

3 - DATA DIVISION: divisão de dados, responsável pela especificação de todas as variáveis e constantes que serão utilizadas dentro do programa;

4 - PROCEDURE DIVISION: divisão de procedimentos. A lógica de seu programa será desenvolvida nesta divisão. Em outras palavras, é exatamente aqui que se inicia seu programa.

As três últimas divisões são sub-divididas em seções, onde em cada uma delas são especificadas informações que, se não forem comentários, serão linhas de comandos que serão executadas pelo programa.

b) Equipamentos e Periféricos

Para rodar um programa em COBOL é necessário possuir um equipamento, como por exemplo, o TK 3000IIe, uma placa de CP/M e uma placa de 80 colunas (que pode ser encaixada no equipamento), dois disk-drives e uma impressora. Seria interessante também uma expansão de memória, isto se forem desenvolvidos programas mais sofisticados ou profissionais.

c) Compilador e LINKeditor

Com os equipamentos necessários, basta apenas montar seu programa com o auxílio de um editor de texto, como por exemplo, o WORDSTAR; depois compilá-lo e "LINKá-lo" pois, para que o computador entenda o programa é necessário que se traduza o programa-fonte para o programa-objeto. Esta é a função do compilador; traduzir, divisão por divisão; apresentar os erros de digitação e posteriormente, unir todas as divisões compiladas. E a função do LINKeditor é deixar o programa pronto para ser executado.

O COBOL não é uma linguagem difícil de ser aprendida, o único problema inicial é ter que decorar suas cláusulas e comandos e saber como utilizá-los, mas com um pouco de prática e dedicação pode-se superar este obstáculo facilmente.

Linguagem C

No início da década de 70, Dennis M. Ritchie desenvolveu a linguagem C com a finalidade de implementar uma das primeiras versões do sistema operacional UNIX. Em pouco tempo, Dennis conseguiu provar, sem grandes dificuldades, o alto padrão da sua criação. Esta linguagem surpreendia a todos, apresentando importantes características de uma das mais bem conceituadas linguagens da época: a linguagem B, que por sua vez teve proveniência de outra famosa linguagem de sistemas, a BCPL.

A BCPL foi criada por Martin Richards especialmente para a área de sistemas e dela vem algumas das mais admiráveis e marcantes características da C, tais como: o estilo direto das expressões e a utilização de pointer (apontadores de variáveis) e arrays (tabelas) conjuntamente.

Na tentativa de aprimorar a BCPL, Ken Thompson desenvolveu a B, cuja maior diferente em relação a C é o fato de somente reconhecer como argumento (variável) códigos de máquinas puros, enquanto que a C aceita os tipos básicos: caracteres, números inteiros curtos e longos, números com precisão dupla e números de ponto-flutuante.

Devido ao que herdou, C se tornou uma linguagem flexível, de propósitos gerais, não sendo específica para nenhuma área e nem se restringindo a determinados processamentos. Com tudo isto, C conseguiu alcançar o título da linguagem mais eficiente, para muitas tarefas.

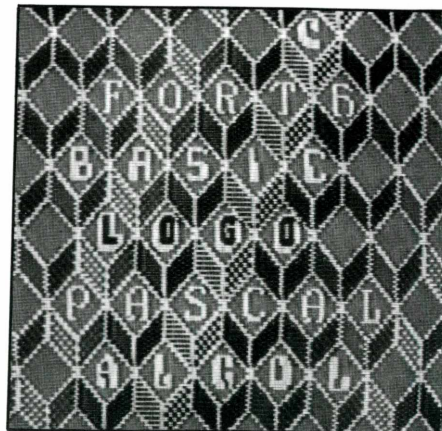
Possuindo características tanto das linguagens de alto-nível quanto das de baixo-nível, ela não se enquadra em nenhum dos dois extremos e é considerada por muitos como uma linguagem de médio-nível que permite a manipulaçao da memória sem com isso tornar-se de difícil aprendizagem.

Os programas em C, desde que escritos na mesma versão da linguagem, podem ser facilmente adaptados para equipamentos diferentes.

Alguém que queira aprender como programar em C deve começar tentando se familiarizar com os conceitos básicos da linguagem, como: tipos de variáveis, utilização e definição de funções, estruturas de controle e expressões de entrada e saída.

Um programador familiarizado com C resolve complexos problemas com programas simples, desde que os recursos da linguagem não sejam trocados por vícios de programação, característicos de programadores de linguagens de alto-nível.

C é uma linguagem estruturada, suas estruturas de controle foram herdadas da BCPL, que por sua vez é uma das mães da programação estruturada. As principais estruturas de controle que C oferece são: IF, para decisões; WHILE e FOR, fazem ***loopings*** com testes no começo ou final do conjunto de comandos; e do SWITCH, usado para selecionar um caso entre vários.

Os programas em C, normalmente, seguem a seguinte linha: um programa principal acessa funções separadas. Pode-se dizer que todos os programas desenvolvidos em C são constituídos de funções que pertencem a sua biblioteca ou que foram definidos pelo programador dentro do programa.

As variáveis podem ser internas ou externas. O primeiro tipo faz com que ela exista somente dentro de determinada função. O segundo tipo pode ser reconhecido pelo programa como arquivo fonte ou como variável global pertencendo deste modo, a todo o programa e não a uma única função.

Como qualquer outra linguagem, C possui defeitos: alguns dos seus operadores assumem uma hierarquia errada; sua sintaxe poderia ser melhor e suas versões apresentam algumas diferenças.

Finalmente, C pode ser considerada uma linguagem do futuro que está ao alcance da maioria dos micros brasileiros. Pode-se afirmar que ela é expressiva e versátil para uma larga variedade de programas, sendo também fácil de aprender e de usar.

Logo

O Logo é mais uma linguagem de programação, ou seja, um meio de comunicação entre o usuário e a máquina. É uma linguagem simples por ser de fácil assimilação e também muito poderosa por apresentar recursos sofisticados. Ela vem sendo considerada a linguagem ideal para ser usada por educadores em um ambiente de aprendizado, devido principalmente, aos seus recursos gráficos.

Esta linguagem pode ser ensinada para pessoas de qualquer idade não havendo a necessidade de conhecimentos de matemática e nem de programação.

Seu aprendizado é muito simples, divertido, e ao mesmo tempo permite, aos mais experientes, a realização de programas bem sofisticados. ◇

Apesar de ser fácil as crianças programarem em Logo, não significa que ela seja uma linguagem infantil.

O Logo apresenta uma estrutura procedural, isto significa que podemos construir procedimentos, formados de uma ou mais instruções, podendo ser comandos primitivos do próprio Logo ou outros procedimentos.

Tanto os comandos primitivos do Logo quanto os procedimentos criados pelo usuário, são utilizados da mesma forma não havendo uma discriminação.

Por ser uma linguagem flexível, esta permite que você crie ou modifique os procedimentos sem maiores dificuldades.

A parte gráfica do Logo produz resultados visuais imediatos, dando um estímulo ao usuário para pesquisar ainda mais a fundo este terreno.

Para que o Logo execute as instruções fornecidas pelo programador, estas devem estar numa forma que possa ser reconhecida pelo Logo.

Quando usamos esta linguagem somente para escrever, dizemos que estamos usando a "tela de texto" No caso de estarmos desenhando estaremos utilizando a "tela de desenho".

Para ser feito qualquer tipo de figura nesta linguagem, é necessário que o usuário forneça todas as informações a respeito do desenho que se pretende executar.

Depois que um procedimento está definido, ele pode ser usado como qualquer outro comando, no modo direto ou fazendo parte de outros procedimentos, chamamos isto de "subprocedimentos".

Pascal

O desenvolvimento da linguagem Pascal foi baseado em dois objetivos principais:

1. que esta fosse uma linguagem própria, e que no ensino de programação se comportasse como uma matéria sistemática, baseando-se em alguns conceitos fundamentais;
2. que suas implementações, fossem confiáveis e eficientes.

O resultado foi uma linguagem amplamente estruturada, com poucos conceitos fundamentais, sendo assim, fácil de aprender.

As características mais importantes do Pascal são: a grande facilidade na estruturação dos dados; a possibilidade do programador definir seus próprios tipos de escala, além dos tipos-padrão oferecidos pela linguagem; e um completo conjunto de comandos estruturados.

Os comandos da linguagem Pascal são classificados em dois tipos básicos:

a. comandos simples
b. comandos estruturados

Em geral, um programa de computador consiste em duas partes:
- uma seção de definição;
- uma seção de instrução.

Na maioria das linguagens de computador, a seçao definição é misturada com a seção de instrução, o que pode dar a impressão de que o programa consiste de uma longa lista de instruções.

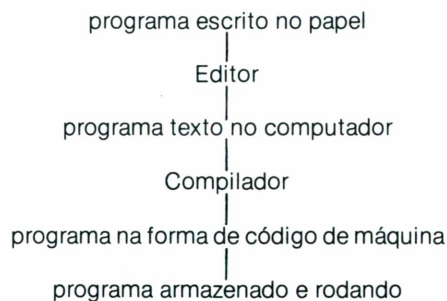
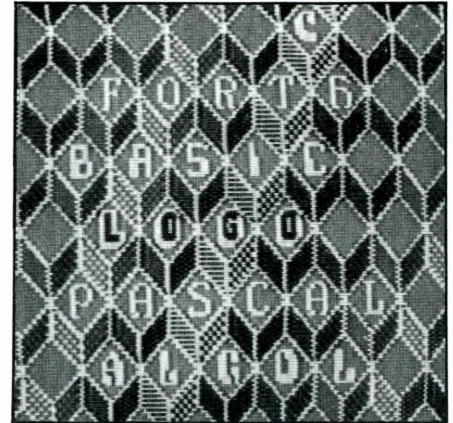
Pascal, contudo, separa as duas seções muito claramente e isto é uma característica importante de qualquer programa escrito em ***Pascal.***

Todos os programas Pascal tomam a seguinte forma:

```
PROGRAM nome
     definições
BEGIN
     instruções
END
```

Esta é uma linguagem de alto nível, e antes que um programa escrito em Pascal possa rodar em um computador, ele deve ser traduzido para código de máquina. Isto é, um programa Pascal envolve ao menos três operações:
- o programa deve ser introduzido no computador como texto;
- deve ser compilado para código de máquina;
- rodado posteriormente, com o computador passando a obedecer suas instruções.

Podemos visualizar este processo, como:

programa escrito no papel
|
Editor
|
programa texto no computador
|
Compilador
|
programa na forma de código de máquina
|
programa armazenado e rodando

De forma diferente do BASIC, o Pascal não é normalmente fornecido com um padrão característico, na maioria dos microcomputadores. Isto significa que, antes que possa começar a aprender ou usar o Pascal, você deve escolher e comprar uma versão da linguagem, que sirva em sua máquina.

Conclusão

Todas as linguagens de programação existentes possuem defeitos e qualidades, cada uma é específica para uma determinada área e desenvolvida para um determinado equipamento, sendo posteriormente, adaptada para outros.

Quando fazemos opção por uma delas, devemos ter em mente a sua aplicação e se as condições para seu uso são ideais, pois se não forem jamais o programador conseguirá desfrutar de suas vantagens.

Lembre-se: cada caso é um caso e o que é bom para um equipamento ou aplicação pode ser ruim para outro. Por isso analise bem antes de escolher a sua linguagem de programação. ◇

Tente Esta! TK-85

```
 10 INPUT A$
 20 FOR I=1 TO LEN A$
 30 PRINT AT 10,16-LEN
A$/2;A$(I TO );A$
( TO I)
  35 PAUSE 40
  40 NEXT I
  50 GOTO 20
```

CP/M

Há dez anos atuando como Sistema Operacional padrão.

Elaine Kelly Lima

Em 1976, enquanto Wozniak aperfeiçoava o Apple II um outro americano chamado Gary Kildall colocou no mercado um produto que mudaria os conceitos sobre sistemas operacionais para microcomputadores: o CP/M.

O ***Control Program for Microcomputer*** teve sua versão mais rudimentar desenvolvida em 1975 durante um período de trabalho experimental, que Kildall fazia, com a PL/M para a empresa onde era programador, a Microcomputers Application Associates (MAA). Ao verificar o impacto que sua criação estava causando ele pediu demissão da MAA e fundou, no mesmo ano, a Digital Research Inc., que até hoje é responsável pela comercialização do sistema.

Quando foi lançado definitivamente no mercado, o CP/M explorou uma fatia do mercado que até o momento permanecia intacta e mesmo assim, em pouco tempo, Kildall conseguiu transformar seu produto em "padrão" para os micros profissionais, sendo que esta fama mantém-se até hoje graças a sua eficiência e potencialidade.

Atualmente, a maioria dos equipamentos oferecem aos seus usuários, além do próprio sistema operacional, uma versão do CP/M. Porém, para carregá-lo, o micro precisa preencher determinados requisitos e ter a seguinte configuração mínima: memória principal de 48 Kbytes; um microprocessador Z80, 8080, 8086 ou 8088; um terminal (teclado e monitor); e um drive para disquete de 5 1/4" ou 8 polegadas.

Apesar dos compatíveis com o Apple processarem sob o 6502, pode-se acoplar uma placa de Z80 que possibilita a utilização do CP/M. Para que essa conexão seja feita é necessário que o micro apresente um slot específico à essa função, tal qual o TK 3000 IIe.

Além de ser um sistema operacional muito flexível e fácil de trabalhar, o CP/M possui uma quantidade incontável de programas compatíveis ou adaptáveis.

Apresentando o CP/M

Antes de mostrarmos o seu modo de operação, vamos fazer uma rápida descrição da sua estrutura interna. Veja a figura 1.

FIGURA 1

Basic INPUT/OUTPUT System (BIOS) Sistema básico de entrada e saída
Basic Disk Operating System (BDOS) Sistema básico de operação de discos
Console Commands Processor (CCP) Processador de comandos da console
Transiente Program Area (TPA) Área de programa transiente
Página 0
ESTRUTURA DA MEMÓRIA SOB O CP/M

Na estrutura mostrada na figura 1, podemos considerar como parte integrante do sistema os BIOS, BDOS e CCP. O primeiro é responsável pela interação software/hardware e sua função é associar os dispositivos lógicos aos físicos. O segundo é quem realiza todas as manipulações com o disquete, tanto para acesso como para organização e alocação de arquivos na unidade física. A CCP é a parte que interpreta os caracteres pressionados pelo usuário no teclado.

A área de programa transiente é o local da memória onde os programas são carregados para que possam ser executados e controlados pelo CP/M. É bom observar que o endereçamento de início da CCP, BDOS e BIOS, depende do tamanho da memória do equipamento utilizado.

Nos micros que aceitam CP/M algumas teclas são fundamentais:

- a tecla que permite executar comandos e ir para a próxima linha pode ser encontrada nos diversos equipamentos com os seguintes nomes: RETURN, ENTER, NEW LINE, CR, etc; neste artigo, ela será indicada por (CR);
- a tecla de controle é fundamental e talvez seja a mais importante. Em nosso artigo ela será representada por ⊡ e é encontrada nos diversos micros com os seguintes nomes: CONTROL, CTRL, CNTL, ALT etc.

Por ser um sistema operacional armazenado em disquete, será carregado na memória quando o equipamento acessar para leitura, um disquete que tenha nas suas duas primeiras trilhas o sistema gravado. E a partir daí, o CP/M passa a controlar todas as operações, sendo que o ***prompt*** apresentado por ele indica o drive ***default*** que normalmente é A> . Isto significa que em todos os comandos cujo drive de acesso não é especificado, o sistema reconhecerá o A: . Caso o usuário queira mudar, temporariamente, o drive ***default***, o procedimento a ser seguido é muito simples:

A > B: <CR>
B >

Como você pode observar, bastou indicar o drive, que neste sistema sempre é uma letra de A até P, seguida de dois pontos.

Trabalhando com o CP/M

Além dos comandos executáveis, estando apenas com o sistema em memória, o CP/M também possui uma grande quantidade de programas transientes que funcionam como utilitários. Esses programas desempenham distintas funções, tais como: copiar, editar e depurar arquivos; formatar e gerar o sistema em disquetes; fornecer informações sobre unidades de entrada e saída (disquetes ou arquivos); exibir no formato hexadecimal o conteúdo de arquivos; converter programas fonte em Assembler para hexadecimal e vice-versa; executar seqüência de comandos do sistema gravados previamente em arquivos; etc.

Ao nomearmos um arquivo neste sistema, devemos prestar atenção à sua sintaxe, pois ele é formado por um prefixo e um sufixo separados por um ponto. O prefixo pode ter no máximo oito caracteres e o sufixo três, sendo que o primeiro representa o nome dado e o segundo indica o tipo. Para formação dos nomes são caracteres inválidos: ⟨ ⟩ .,;: = ?( )[ ] e os caracteres especiais utilizados em comandos são: ? que indica qualquer caractere; e * que indica qualquer string.

Para conhecer os tipos mais comuns de sufixos observe a tabela 1.

TABELA 1

TIPO	REPRESENTAÇÃO
.ASM	Programa fonte em Assembler.
.BAS	Programa fonte em BASIC.
.CMD	Program fonte DBase.
.COB	Programa fonte em COBOL.
.FOR	Programa fonte em Fortran.
.PAS	Programa fonte em Pascal.
.PLI	Programa fonte em PL/I.
.COM	Programa transiente executável.
.REL	Programa relocável em Linguagem de Máquina.
.OBJ	Programa em código objeto.
.$$$	Arquivo temporário, inútil ou mal gravado.

Exemplos de nomes de arquivos:

PROGRAMA.COB ARQUIVO1.TXT
PIP.COM ED.COM

PROG234.ASM MEMOØ2.COM
F88.REL MMDDAA86.CMD

Uma ótima característica apresentada pelo CP/M é a de permitir ao usuário utilizar além dos comandos e programas transientes, uma série de controles que facilitam muito a vida do operador. Para conhecê-los, observe a tabela 2.

TABELA 2

TECLAS	FUNÇÃO
^C	Informa ao equipamento que o disquete foi trocado.
^E	Pula para o início da linha seguinte, permitindo ao operador prosseguir a lista de comandos.
^H	Volta um caractere, apagando-o.
^J	O mesmo que RETURN.
^M	O mesmo que RETURN.
^P	Aciona e desaciona a impressora.
^R	Repete linha corrente.
^S	Provoca parada temporária de display no vídeo e na impressora. Qualquer tecla continua.
^U	Vai para a linha seguinte, cancelando a atual.
^X	Apaga a linha corrente retornando ao seu início.
^Z	Finaliza entrada pela console.
^ indica tecla de controle: CONTRO, CRTL, CNTL, ou ALT.	

***Obs.:** A tecla de controle deve ser pressionada ao mesmo tempo que a letra, para que haja o retorno desejado.*

Agora que já sabemos nomear arquivos, mudar o drive ***default*** e as teclas de controle do CP/M, vamos aos comandos principais que somam um total de seis, e são muito importantes para quem deseja operar o CP/M.

DIR - lista o diretório do disquete, ou seja, exibe o nome dos arquivos.

Formato: DIR (drive:) (nomearq.tip) CR

Sendo que, no formato apresentado, o nome do arquivo e o drive são parâmetros opcionais.

EXEMPLO	FUNÇÃO
A > DIR <CR>	Exibe o nome de todos os arquivos do drive A:
A > DIR B: <CR>	Exibe o nome de todos os arquivos do drive B:, isso ocorre sem que o drive ***default*** seja mudado.
A > B: <CR> B > DIR <CR>	Exibe todos os arquivos do drive B:, deixando como drive ***default*** o B:
B > DIR*.COB <CR>	Exibe o nome de todos os arquivos do drive B: com sufixo .COB.

TYPE - Exibe o conteúdo de arquivo gravado no formato ASCII, no vídeo ou impressora.

FORMATO: TYPE (drive:) (nomearq.tip) (âP) CR.

Sendo que âP deve ser pressionado somente se houver a necessidade de acionar a impressora.

EXEMPLO	FUNÇÃO
A > TYPE B:PROG.TXT <CR>	Exibe no vídeo o conteúdo do arquivo PROG.TXT que está no drive B:
A > TYPE PR2.FORâP <CR>	Exibe no vídeo e lista na impressora o conteúdo do arquivo PR2.FOR que está no drive A:

REN - Renomeia arquivos.

FORMATO: REN (drive:) (nomenovo.tip) = (drive:) (nomevelho.tip) <CR>

Sendo que, se forem especificados drives diferentes será detectado erro.

EXEMPLO	FUNÇÃO
A > REN B:NOVO COB = B: VELHO COB <CR>	Procura no drive B: o arquivo VELHO.COB e muda seu nome para NOVO.COB.

ERA - deleta arquivos do disquete.

FORMATO: ERA (drive:) (nome.tip) <CR>

EXEMPLO	FUNÇÃO
A > ERA B:*.TXT <CR>	Deleta no drive B: todos os arquivos com sufixo .TXT.
A > ERA*.* <CR>	Provoca a pergunta: ALL (Y/N)?, se a resposta for Y o disquete que está no drive A: terá todos os seus arquivos apagados.
A > ERA PROG.* <CR>	Apaga do drive A: todos os arquivos com prefixo .PROG.
A > ERA P???.FOR <CR>	Apaga do A: todos os arquivos que comecem com a letra P, tenham quaisquer dos três caracteres, e cujo sufixo seja .FOR.

SAVE - Grava no disquete o conteúdo da memória a partir do endereço 100H.

FORMATO: SAVE n (drive:) nome.tip CR

Sendo ***n*** o número de blocos de 256 bytes a serem gravados. Neste comando, o nome do arquivo e ***n*** devem ser determinados pelo operador, obrigatoriamente.

EXEMPLO	FUNÇÃO
A > SAVE 4 B:TRAB.COM <CR	Será gravado no arquivo TRAB.COM no B: o bloco de memória entre 100H e 4FFH.
A > SAVE 7 ARQUIVO.COM <CR>	O bloco de memória entre 100H e 799H é gravado no A:ARQUIVO.COM.

USER - Altera o número do usuário corrente.

FORMATO: USER ***n***

Sendo ***n*** um número inteiro entre 0 e 15 inclusive, o ***default*** é Ø.

EXEMPLO	FUNÇÃO
A > USER 5 <CR>	O usuário corrente agora é o 5.

A partir deste artigo traremos, em todas edições, futuras dicas e usos para os programas transientes desse fantástico sistema operacional. ◇

PROGRAMAS

UTILITÁRIO

P. 012

Editor Hexadecimal para o TK-85

Dirceu Teixeira

Todos nós, usuários do TK-85, sabemos como é cansativo o processo de introdução de programas em Linguagem de Máquina: sempre tendo que digitar um programa em BASIC para a introdução dos códigos em decimal ou em hexadecimal.

Foi pensando muito nisto que desenvolvi este programa, visando facilitar a digitação dos programas em forma de códigos hexadecimais. Como o próprio nome diz, ele trabalha em hexadecimal, mas como a maioria dos programas sempre apresenta o endereço inicial em decimal, mostrarei, logo mais, como converter os endereços em decimal para hexadecimal.

Agora, darei uma explicação dos comandos do editor.

comando "B" - este comando tem, como função, abandonar o editor, retornando ao BASIC.

comando "C" - sua função é apenas a de enviar à impressora uma cópia do que estiver na tela do micro. Para aqueles que não possuem impressora o comando fará apenas a imagem pular.

comando "E" - este é o comando principal do editor e sua função é de dar entrada aos códigos hexadecimais. Ao contrário dos programas em BASIC, não será necessário digitar NEW LINE após cada código, pois o comando faz isso automaticamente.

No início do comando lhe será solicitado o endereço inicial do programa. Feito isto, logo em seguida o programa pede a confirmação do endereço, bastando para isso, digitar "S" ou "N". Confirmado, o sistema imprimirá o endereço em hexadecimal e, a partir daí, é só digitar os códigos.

O comando formata cada linha com sete bytes e a soma sintática de cada linha. Na última linha, o comando imprime a seguinte mensagem: "pressione STOP para menu".

Teclando-se STOP o comando é "abortado", imprimindo a soma sintática da linha digitada. Caso você tenha digitado meio byte (apenas um dígito hexadecimal) esta instrução será ignorada.

Se você digitar um código errado, utilize os seguintes processos para correção:

1. tome nota do endereço da linha, para correção no final do programa e digite toda a linha novamente.
2. tecle STOP para retornar ao menu, entre novamente no comando "E", fornecendo, como endereço inicial, a linha que apresenta o erro e reinicie a digitação a partir daí.

comando "L" - este comando fará uma listagem de um bloco da memória, digitando-se o endereço inicial e a quantidade de bytes a serem listados; a listagem será apresentada no mesmo formato do comando "E".

No caso de se digitar a quantidade de bytes errados, você deve interromper a impressão, pressionando novamente a tecla "L" para continuar listando, ou então, retorne ao BASIC e digite POKE 16507,0 e em seguida, volte ao editor e recomece novamente sua listagem, fornecendo RAND URS 30726, NEW LINE.

comando "R" - este é o último e mais poderoso dos comandos, sua função é criar uma linha REM de tantos bytes quantos forem informados no início do mesmo. Este comando possui um sistema de proteção de linhas muito poderoso, veja como esta proteção funciona; nós sabemos que, quando temos uma linha com mais de 700 bytes e a primeira linha imediatamente abaixo desta for deletada, o micro entra num sistema de "loop" infinito que só parará com o desligamento do equipamento.

Esta proteção cria uma outra linha REM, quando o tamanho da primeira for maior que 512 bytes, com a seguinte mensagem: "não delete esta linha".

Estes são os cinco comandos utilizados pelo programa.

Conversão de endereço decimal para hexadecimal

1. devemos "quebrar" o endereço em dois bytes, dividindo-o por 256. Pegando a parte inteira obteremos o byte mais significativo do endereço;
2. multiplicaremos este valor por 256 e o resultado devemos subtrair do endereço fornecido inicialmente obtendo assim, o byte menos significativo.

Veja como seria, na prática:

16514/256 = 64 (byte mais significativo)
64*256 = 16384
16514-16384 = 130 (byte menos significativo)

Para conferir se a operação está correta, basta fazer o seguinte:
130 + 256*64 isto deverá ser igual a 16514

Para obter o endereço em hexadecimal devemos fazer, agora, a conversão de cada um desses números da seguinte maneira:

1. dividir os números por 16 e pegar somente a parte inteira:

64/16 = 4 (1º dígito)
64-(4*16) = 0 (2º dígito)
130/16 = 8 (3º dígito)
130-(8*16) = 2 (4º dígito)

O resultado em hexadecimal será 4082.

Prestando bastante atenção, o processo é igual ao descrito anteriormente, a única diferença é que, em vez de dividirmos por 256, nós dividiremos por 16, que é a base hexadecimal.

Caso os dígitos estejam entre 10 e 15 estes valores serão substituídos por letras, conforme o esquema abaixo:

10 = A
11 = B
12 = C
13 = D
14 = E
15 = F

Para fixar bem o processo de conversão, vamos a um outro endereço que use letras.

Vamos converter o endereço 18987:

1º passo 18987/256 = 74 (parte inteira)
2º passo 18987-256*74 = 43
3º passo 74/16 = 4 (parte inteira)
4º passo 74-(16*4) = 10 = A (conforme esquema)
5º passo 43/16 = 2 (parte inteira)
6º passo 43-(16*2) = 11 = B (conforme esquema)

O resultado final será 4A2B.

Bem, terminada as explicações de como converter endereços decimais para hexadecimais, vamos para o programa. Afinal, este é o objetivo do artigo.

Crie uma linha REM de 1.085 bytes. Para isto, utilize o programa gerador de linha REM, publicado na edição número 7 da Microhobby.

Inicialmente, digite o programa 1 (a linha REM deve ter 96 pontos).

Tecle RUN e digite os números decimais da listagem 1.

Depois de digitar todos os códigos, peça ao computador para mostrar a listagem. Por meio de um LIST, esta deverá estar conforme o programa 2.

Retire as linhas 2 a 8 e digite as novas linhas do programa 3.

Rode o programa, a seguir digite o comprimento que a sua linha REM deveria ter.

Depois de criada a linha REM, "delete" a linha 1 e a linha 4 em diante. Logo após, digite POKE 16510,0 e 1 REM e a seguir "delete" a linha 3.

Agora digite o programa 4, que fará a entrada dos endereços da listagem 2.

Digite apenas sete bytes por vez, separados por um espaço. Após a apresentação de cada linha será apresentada a soma sintática, onde você poderá conferir com a soma da listagem. No caso da soma estar errada, digite "ERRO" interrom-

pendo o programa, e reedite a linha. Depois de tudo "delete" todo o programa, menos a linha 1, e digite o programa 5, gravando antes de executar o mesmo.

Para rodar o programa, você deve dar um RAND USR 30726, porque este foi desenvolvido para trabalhar acima da RAM TOP.

```
   1 REM ...........................
................................
................................
.......
   2 FOR I=16514 TO 16609
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT X
   6 PRINT X
   7 POKE I,X
   8 NEXT I
```

```
   1 REM   LN 7?▘▗▘ E▙RND FAST ▒?
?EDRND▒6DRND5£RNDY▒?7?STR$ FOR ▒
 FOR ?F?77XC▀SGN / INPUT LPRINT
FAST ▘ LLIST RND▩ GOSUB PI?? LPR
INT GOSUB ▩5 STOP RNDQ 70 ▘7AT ▀
??77Q REM ""7)▘  FOR ; FOR Q. GO
SUB ▩Q?OLN F?TAN
   2 FOR I=16514 TO 16609
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT X
   6 PRINT X
   7 POKE I,X
   8 NEXT I
```

```
   1 REM * LN 7?▘▗▘ E▙RND FAST ▒?
?EDRND▒6DRND5£RNDY▒?7?STR$ FOR ▒
 FOR ?F?77XC▀SGN / INPUT LPRINT
FAST ▘ LLIST RND▩ GOSUB PI?? LPR
INT GOSUB ▩5 STOP RNDQ 70 ▘7AT ▀
??77Q REM ""7)▘  FOR ; FOR Q. GO
SUB ▩Q?OLN F?TAN
   3 PRINT "COMPRIMENTO DA REM:"
   4 INPUT C
   5 PRINT C
   6 PAUSE 60
   7 POKE 16514,C-256*INT (C/256
)
   8 POKE 16515,INT (C/256)
   9 RAND USR 16516
  10 CLS
  11 LIST 2
```

```
  10 LET A$=""
  20 LET EN=16514
  30 IF A$="" THEN LET SS=0
  40 IF A$="" THEN SCROLL
  50 IF A$="" THEN PRINT EN;" ";
  60 IF A$="" THEN INPUT A$
  70 IF A$="ERRO" THEN GOTO 150
  80 LET X=16*CODE A$(1)+CODE A$
(2)-476
  90 PRINT A$( TO 2);" ";
 100 POKE EN,X
 105 LET SS=SS+X
 110 LET A$=A$(4 TO )
 120 LET EN=EN+1
 130 IF A$="" THEN PRINT " ";SS
 140 GOTO 30
 150 LET A$=""
 160 LET EN=EN-7
 170 GOTO 30
```

```
   2 POKE 16388,255
   3 POKE 16389,119
   4 FOR N=0 TO 1199
   5 POKE (30720+N),PEEK (16514+
N)
   6 NEXT N
   7 PRINT AT 10,11;"TUDO OK:"
   8 PAUSE 500
   9 NEW
```

```
16514..0     16546..9     16578..54
16515..0     16547..235   16579..2
16516..205   16548..114   16580..35
16517..35    16549..43    16581..193
16518..15    16550..115   16582..3
16519..1     16551..35    16583..3
16520..6     16552..35    16584..113
16521..0     16553..61    16585..35
16522..42    16554..40    16586..112
16523..130   16555..3     16587..35
16524..64    16556..209   16588..54
16525..229   16557..24    16589..234
16526..9     16558..238   16590..11
16527..68    16559..225   16591..11
16528..77    16560..229   16592..35
16529..42    16561..1     16593..17
16530..41    16562..226   16594..1
16531..64    16563..64    16595..0
16532..9     16564..167   16596..235
16533..34    16565..237   16597..25
16534..41    16566..66    16598..235
16535..64    16567..68    16599..54
16536..33    16568..77    16600..27
16537..12    16569..225   16601..237
16538..64    16570..237   16602..176
16539..62    16571..184   16603..54
16540..9     16572..33    16604..117
16541..94    16573..227   16605..52
16542..35    16574..64    16606..205
16543..86    16575..54    16607..43
16544..213   16576..0     16608..15
16545..235   16577..35    16609..201
```

PROGRAMAS

```
4082  00 00 00 00 00 00 CD  205
4089  67 7A CD 38 7B CD 7B  937
4090  7B 3A 21 40 FE 27 C8  771
4097  FE 28 CC 69 08 FE 31  914
409E  CA B6 78 FE 37 CA 6A  1121
40A5  79 FE 2A 20 E4 CD 0E  896
40AC  0C CD 67 7A CD AE 7A  943
40B3  21 01 78 E5 CD 92 7B  857
40BA  E1 77 2B E5 CD 92 7B  1090
40C1  E1 77 CD 67 7A CD CF  1186
40C8  7A 2A 00 78 CD D6 7B  826
40CF  CD E9 7A CD 7B 7B 3A  1069
40D6  21 40 FE 33 28 D1 FE  905
40DD  38 20 F2 CD 67 7A CD  965
40E4  F6 7A 21 00 00 22 04  439
40EB  78 CD 0E 0C 2A 00 78  513
40F2  CD D6 7B D7 06 07 C5  967
40F9  CD 92 7B F5 3A 21 40  874
4100  FE E3 28 22 F1 2A 00  838
4107  78 77 23 22 00 78 06  434
410E  00 4F 2A 04 78 09 22  288
4115  04 78 AF D7 C1 10 DC  943
411C  D7 ED 4B 04 78 CD 98  1008
4123  0A 18 C0 F1 C1 01 1C  689
412A  15 CD F5 08 ED 4B 04  795
4131  78 CD 98 0A C3 06 78  808
4138  3A 7B 40 FE 31 28 38  644
413F  CD 67 7A CD AE 7A 21  964
4146  01 78 E5 CD 92 7B E1  1049
414D  77 2B E5 CD 92 7B E1  1090
4154  77 CD 67 7A CD 17 7B  900
415B  21 03 78 E5 CD 92 7B  859
4162  E1 77 2B E5 CD 92 7B  1090
4169  E1 77 AF 2A 00 78 ED  918
4170  5B 02 78 19 22 02 78  394
4177  CD 2A 0A 3E 16 F5 2A  628
417E  00 78 CD D6 7B D7 06  883
4185  07 C5 2A 00 78 7E 06  498
418C  00 4F 2A 04 78 09 22  288
4193  04 78 2A 00 78 7E CD  617
419A  DC 7B 23 22 00 78 ED  769
41A1  5B 02 78 ED 52 28 22  606
41A8  C1 10 DB AF D7 ED 4B  1130
41AF  04 78 CD 98 0A 01 00  492
41B6  00 ED 43 04 78 3E 76  608
41BD  D7 F1 3D 20 BA 3E 31  846
41C4  32 7B 40 C3 06 78 C1  751
41CB  F1 4F 3E 16 91 47 0E  634
41D2  1C CD F5 08 ED 4B 04  802
41D9  78 CD 98 0A 01 00 00  488
41E0  ED 43 04 78 3E 00 32  540
41E7  7B 40 C3 06 78 CD 67  816
```

```
41EE  7A CD 17 7B 21 03 78  629
41F5  E5 CD 92 7B E1 77 2B  1090
41FC  E5 CD 92 7B E1 77 CD  1252
4203  23 0F 01 06 00 2A 02  101
420A  78 E5 09 44 4D 2A 29  586
4211  40 09 22 29 40 21 0C  257
4218  40 3E 09 5E 23 56 D5  563
421F  EB 09 EB 72 2B 73 23  786
4226  23 3D 28 03 D1 18 EE  610
422D  E1 E5 01 7C 40 A7 ED  1047
4234  42 44 4D E1 ED B8 21  890
423B  7D 40 36 00 23 36 00  332
4242  23 C1 03 03 71 23 70  494
4249  23 36 EA 0B 0B 23 11  397
4250  01 00 EB 19 EB 36 1C  578
4257  ED B0 36 75 34 3A 03  697
425E  78 FE 02 DA F5 7B 22  996
4265  14 7A 23 22 1F 7A 01  365
426C  06 00 21 17 00 E5 09  300
4273  44 4D 2A 29 40 09 22  335
427A  29 40 21 0C 40 3E 09  285
4281  5E 23 56 D5 EB 09 EB  907
4288  72 2B 73 23 23 3D 28  443
428F  03 D1 18 EE E1 E5 01  929
4296  D7 44 A7 ED 42 44 4D  898
429D  E1 ED B8 21 D8 44 36  1017
42A4  00 23 36 01 23 C1 03  321
42AB  03 71 23 70 23 36 EA  586
42B2  0B 0B 23 11 01 00 EB  310
42B9  19 EB 36 00 ED B0 36  781
42C0  75 34 2B 11 66 7A EB  688
42C7  01 17 00 ED B8 CD 2B  693
42CE  0F C3 F8 7B 80 B3 A6  1054
42D5  B4 80 A9 AA B1 AA B9  1179
42DC  AA 80 AA B8 B9 A6 80  1131
42E3  B1 AE B3 AD A6 80 CD  1202
42EA  5D 7B 16 00 AA A9 AE  751
42F1  B9 B4 B7 80 AD AA BD  1208
42F8  A6 A9 AA A8 AE B2 A6  1191
42FF  B1 80 B9 B0 98 A4 A1  1143
4306  80 90 A8 91 9D A5 A4  1071
430D  A2 05 00 00 00 00 00  167
4314  00 00 00 00 00 00 00  0
431B  00 00 00 00 00 00 00  0
4322  00 00 00 00 00 00 00  0
4329  00 00 00 00 85 FF C9  589
4330  CD 5D 7B 17 01 29 2E  532
4337  2C 2E 39 2A 00 2A 33  282
433E  29 2A 37 2A 28 34 00  272
4345  2E 33 2E 28 2E 26 31  316
434C  00 12 00 FF C9 CD 5D  772
4353  7B 17 01 28 34 33 2B  333
```

```
435A  2A 37 2A 00 2A 33 29  273
4361  2A 37 2A 28 34 00 12  249
4368  00 FF C9 CD 5D 7B 17  900
436F  19 10 38 18 33 11 0F  204
4376  FF C9 CD 5D 7B 17 03  903
437D  35 37 2A 38 38 2E 34  360
4384  33 2A 00 0B 38 39 34  269
438B  35 0B 00 35 26 37 26  248
4392  00 32 2A 33 3A FF C9  657
4399  CD 5D 7B 17 01 29 2E  532
43A0  2C 2E 39 2A 00 34 00  241
43A7  33 3A 32 2A 37 34 00  308
43AE  29 2A 00 27 3E 39 2A  283
43B5  38 0E 00 FF C9 CD 5D  824
43BC  7B 17 01 32 2A 33 3A  348
43C3  00 A7 26 38 2E 28 00  347
43CA  A8 34 35 3E 00 AA 33  556
43D1  39 2A 37 00 B1 2E 38  433
43D8  39 00 B7 2A 32 FF C9  788
43DF  AF 32 22 40 E1 46 23  653
43E6  4E 23 E5 CD F5 08 E1  1025
43ED  7E 23 E5 FE FF 28 03  942
43F4  D7 18 F5 3E 02 32 22  632
43FB  40 C9 CD BB 02 2C 20  735
4402  FA CD BB 02 2C 28 FA  978
4409  2D E5 C1 CD BD 07 7E  994
4410  32 21 40 C9 01 00 00  349
4417  ED 43 7B 40 CD 7B 7B  942
441E  3A 21 40 FE E3 20 08  676
4425  3A 7B 40 FE 00 20 EF  770
442C  C9 FE 1C 38 EA FE 2C  1071
4433  30 E6 D7 C6 E4 4F 3A  1056
443A  7B 40 A7 20 14 06 04  416
4441  CB 21 10 FC 79 32 7C  799
4448  40 3A 7B 40 C6 80 32  685
444F  7B 40 18 C8 3A 7C 40  657
4456  81 C9 7C 4F CD E3 7B  1088
445D  7D 4F CD E3 7B AF D7  1149
4464  C9 E6 F0 06 04 CB 3F  947
446B  10 FC C6 1C D7 3E 0F  786
4472  A1 C6 1C D7 C9 CD 2B  1051
4479  0F CD 5D 7B 17 01 34  512
4480  30 1A 00 00 31 2E 33  220
4487  2D 26 00 80 B7 AA B2  742
448E  80 00 29 2A 00 00 00  211
4495  00 00 00 27 3E 39 2A  200
449C  38 FF 3E 00 32 22 40  521
44A3  01 15 17 CD F5 08 2A  545
44AA  02 78 CD D6 7B 01 FF  920
44B1  FF 0B 78 B1 20 FB 3E  908
44B8  02 32 22 40 C3 06 78  471
```

Desenhando com o teclado

Solte sua criatividade e divirta-se fazendo figuras e gráficos no **TK 2000.**

Fernando Carvalho Filho

Este programa, desenvolvido para o TK-2000, torna o vídeo do seu equipamento uma tela reservada a desenhos que serão feitos por você sem a utilização de lápis ou pincéis, mas através do teclado ou do joystick. Além de explorar a criatividade do usuário divertindo-o, o programa poderá revelar em você, uma aptidão muito especial e às vezes, desconhecida.

Através da apresentação de um menu inicial, o programa informa quais são as teclas de controle. Para traçar pontos que sobem, descem, que vão para direita ou ainda para esquerda deve-se pressionar as setas do teclado, respectivas ao movimento desejado. Para desenhar na diagonal use:

A - esquerda para baixo;
S - direita para baixo;
Q - esquerda subindo;
W - direita subindo.

Se por acaso alguma linha for desenhada de forma erra, você poderá apagá-la pressionando a letra ***C***. Após pressionar esta letra será necessário passar novamente sobre as posições que você deseja apagar e, desta maneira, obter o satisfatório. Para prosseguir com o desenho, pressione novamente ***C***, caso contrário, o HCOLOR continuará igual a zero e nada será exibido.

O programa oferece ainda, rotinas especiais para leitura e gravação da figura. Além disto, ao estar no modo gráfico, você poderá retornar ao menu sem perder o desenho, através das teclas: ***M*** (vai para o menu) e ***D*** (vai para o desenho).

Caso você queira parar o programa, o desenho será perdido. Para voltar a rodá-lo basta pressionar as duas teclas RESET juntas, já que o programa ao ser rodado pela primeira vez, joga na memória do equipamento uma rotina de máquina que força essa situação.

Para facilitar o entendimento da estrutura do programa colocamos a seguir, uma tabela de suas rotinas:

Tabela de Rotinas

Linhas	Função
6 a 8	Rotina jogada na memória para provocar RUN automático, ao serem pressionadas as teclas RESET.
9 a 10	Prepara a tela para o desenho.
12 a 50	Apresenta o menu.
55 a 105	Leitura do teclado.
125 a 250	Endereço das rotinas e valores do HPLOT.
500 a 650	Consistência dos valores do HPLOT.
1000 a 1020	Exibição na tela.
1200 a 1220	Limitações do HPLOT.
1500 a 1560	Define a cor do ponto.
2000 a 2550	Fim do programa.
3000	Prepara rotinas de leitura e gravação de desenhos.
4000 a 4080	Rotina de gravação de figuras.
5000 a 5080	Rotina de leitura de figuras.
6000 a 6030	Entrada do nome do desenho.

```
5  REM  DESENHANDO
6  DATA 169,123,141,242,3
,169,200,141,
243,3,32,116,251,96
7  FOR I = 768 TO 781: RE
AD BYTE: POKE
I,BYTE: NEXT
8  CALL 768
9  MA : TEXT : MP : HGR2
: COLOR  = 3
10  FOR I = 0 TO 45: HLIN
 0,39 AT I: NEXT
12  MA
15  VTAB 1: HTAB 14: PRIN
T "B12/06/198
6F"
20  VTAB 3: HTAB 12: INVE
RSE : PRINT "
FAZENDO DESENHOS": NORMAL

25 A = 139:B = 96:F = 3:C
 = 0
35  VTAB 5: PRINT "*MENU*
"
37  PRINT "(M)---)MENU"
38  PRINT "(D)---)DESENHO
"
39  PRINT "(C)---)APAGA A
 LINHA DESENH
ADA"
40  PRINT "(A)---)DIAGONA
L ESQUERDA P/
 BAIXO"
41  PRINT "(S)---)DIAGONA
L DIREITA P/
BAIXO"
42  PRINT "(Q)---)DIAGONA
L ESQUERDA P/
 CIMA"
43  PRINT "(W)---)DIAGONA
L DIREITA P/
CIMA"
44  PRINT "((-)--)ESQUERD
A"
45  PRINT "(-))--)DIREITA
"
46  PRINT "(SETA P/ CIMA)
---)VERTICAL
P/ CIMA"
47  PRINT "(SETA P/ BAIXO
)--)VERTICAL
P/ BAIXO"
48  PRINT "(L)---)LER DES
ENHO"
49  PRINT "(G)---)GRAVAR
DESENHO"
50  PRINT "(F)---)SAIR FO
RA DO PROGRAM
A"
55  VTAB 23: HTAB 15: INV
ERSE : PRINT
"DIVIRTA-SE!": NORMAL
60  HCOLOR  = C
100 H =  PEEK (39)
101  PRINT "HPLOT "A","B
105  IF H = 48 GOTO 100
125  IF H = 8 THEN  MA :
GOTO 2500
130  IF H = 9 THEN  MP :
GOTO 100
135  IF H = 7 THEN  MA :
GOTO 4000
140  IF H = 44 THEN  MA :
 GOTO 100
145  IF H = 40 THEN  MA :
 GOTO 5000
150  IF H = 3 GOTO 2000
155  MP
160  IF H = 24 THEN  GOSU
B 500:A = A +
1: GOTO 1000
170  IF H = 18 THEN  GOSU
B 520:A = A -
1: GOTO 1000
180  IF H = 36 THEN  GOSU
B 540:B = B -
1: GOTO 1000
190  IF H = 30 THEN  GOSU
B 560:B = B +
1: GOTO 1000
200  IF H = 16 THEN  GOSU
B 580:A = A +
1:B = B - 1: GOTO 1000
210  IF H = 17 THEN  GOSU
B 600:A = A -
1:B = B - 1: GOTO 1000
220  IF H = 11 THEN  GOSU
B 620:A = A -
1:B = B + 1: GOTO 1000
230  IF H = 10 THEN  GOSU
```

PROGRAMAS

```
B 640:A = A +
1:B = B + 1: GOTO 1000
250  GOTO 100
500  IF A = 277 GOTO 1200

510  RETURN
520  IF A = 1 GOTO 1200
530  RETURN
540  IF B = 1 GOTO 1200
550  RETURN
560  IF B = 182 GOTO 1200

570  RETURN
580  IF A = 277 OR B = 1
GOTO 1200
590  RETURN
600  IF A = 1 OR B = 1 GO
TO 1200
610  RETURN
620  IF A = 1 OR B = 182
GOTO 1200
630  RETURN
640  IF A = 277 OR B = 18
2 GOTO 1200
650  RETURN
1000  IF F = C GOTO 1500
1010  HPLOT A,B
```

```
1020  GOTO 100
1200  FOR I = 1 TO 3
1210  PRINT  CHR$ (7)
1220  NEXT
1230  GOTO 100
1500  IF F = 3 THEN S = 0

1510  IF F = 0 THEN S = 3

1520  HCOLOR  = S
1530  HPLOT A,B
1540  HCOLOR  = 3
1550  HPLOT A,B
1560  GOTO 100
2000  PRINT  CHR$ (7)
2010  IF C = 3 THEN C = 0
: HCOLOR  = C
: HPLOT A,B: GOTO 100
2020 C = 3
2030  HCOLOR  = C
2040  HPLOT A,B
2050  GOTO 100
2500  VTAB 24: PRINT "CON
FIRMA FINAL (
S/N)?";
2510  GET H$
2520  IF H$ <  > "S" THEN
 100
```

```
2530  TEXT
2540  CALL 64351
2550  END
3000  POKE 61,160: POKE 6
0,00: POKE 63
,191: POKE 62,255: RETURN

4000  VTAB 24: PRINT "CON
FIRMA GRAVACA
O (S/N)?";: GET H$
4010  IF H$ <  > "S" THEN
 100
4020  PRINT : INPUT "NOME
 DO DESENHO (
6)---)";H$
4030  IF  LEN (H$) > 6 TH
EN 4020
4040  FOR I =  LEN (H$) +
 1 TO 6:H$ =
H$ + " ": NEXT
4050  GOSUB 6000: GOSUB 3
000
4060  PRINT : PRINT "AGUA
RDE UNS INSTA
NTES...";
4070  CALL 60570
4080  GOTO 100
```

```
5000  VTAB 24: PRINT "CON
FIRMA LEITURA
 (S/N)?";: GET H$
5010  IF H$ <  > "S" THEN
 100
5020  PRINT : INPUT "NOME
 DO ARQUIVO (
6 LETRAS)-)";H$
5030  IF  LEN (H$) <  > 6
 THEN 5020
5040  FOR I =  LEN (H$) +
 1 TO 6:H$ =
H$ + " ": NEXT
5050  GOSUB 6000: GOSUB 3
000
5060  PRINT : PRINT " AGU
ARDE...";
5070  CALL 60846
5080  GOTO 100
6000  FOR I = 1120 TO 112
5
6010  POKE I, ASC ( MID$
(H$,I - 1119,
1)) + 128
6020  NEXT
6030  RETURN
```

◇

Invasores 1 - TK-85

JOGO

Paulo Panagiotis Alvanos

Mais um divertido jogo entra para sua coleção. "Invasores" é uma micro réplica do famoso jogo Space Invaders, que tem por finalidade defender sua base dos alienígenas que pousam, lentamente, sobre a superfície de seu planeta. Para evitar que isso aconteça é necessário atacá-los com seus tiros até que eles recuem.

Sua digitação é simples não oferecendo maiores problemas.

Para movimentação de sua base utilize as teclas "5" para esquerda, "8" para a direita e "V" para atirar. Ao final do jogo é apresentado sua pontuação e se deseja continuar ou não. Bom jogo...

```
  10 CLS
  15 LET P=8
  20 LET D=P=P
  30 LET N=-D
  40 LET Y=N-N
  50 LET N=N+1
  60 LET X=INT (RND*7)
  70 LET C=N-N
  80 LET Y=Y+1
  90 IF X=1 OR X=9 OR RND>.7 THE
N LET D=-D
 100 LET X=X+D
 110 IF Y>15 THEN PRINT "**BASE
INVADIDA...PONTOS=";N
 115 IF Y>15 THEN GOTO 210
 120 SCROLL
 130 LET P=P-(INKEY$="5" AND P)+
(INKEY$="8" AND P<10)
```

```
 140 PRINT AT 15,P;"  ";"■ " AND
INKEY$="V";"  "
 150 PRINT AT 16,P;" ▙ ";AT Y,X
;
 160 IF PEEK (PEEK 16398+PEEK 16
399*256)=1 THEN GOTO 40
 170 PRINT "*"
 180 LET C=C+1
 190 IF C>=4 THEN GOTO 70
 200 GOTO 120
 210 CLS
 220 PRINT AT 10,3;"CONTINUA(S/N
)"
 230 IF INKEY$="S" THEN GOTO 10
 235 IF INKEY$="N" THEN GOTO 250
 240 GOTO 230
 250 STOP
```

◇

Pelicano

Mergulhe nesta aventura com seu **TK-90X.**

Elaine Kelly Lima

Como todos sabemos, o pelicano é uma ave aquática de pernas curtas que, para sobreviver, ocupa-se da pesca. Sendo um hábil pescador, ele espera o momento exato para mergulhar e apanhar, de improviso, seu alimento.

Neste joguinho desenvolvido para o TK-90X com RAM de 16 ou 48 Kbytes, você terá a oportunidade de simular um pelicano que, para sobreviver, ataca os habitantes do mundo submarino.

A manipulação do pelicano pode ser feita com joystick ou através do próprio teclado. No segundo caso, as teclas de controle são:

5 - muda o vôo para a esquerda;
6 - faz com que a ave abaixe o vôo;
7 - a ave voa mais alto;
8 - o sentido do vôo é para a direita;
9 - provoca mergulho do pelicano.

Para passar da primeira fase a ave, controlada por você, terá que comer 7 peixes, sendo que, para isso, não poderá ultrapassar o tempo especificado pelo programa. Caso não consiga, seu pelicano perderá o vigor e por conseqüência, morrerá.

A cada nível o tempo especificado para completar a pescaria diminui e deste modo, somente a sua habilidade como pescador poderá mantê-lo vivo.

Sub-rotinas do programa

A linha 10 inicializa a tela e chama duas sub-rotinas: linha 1050 que forma caracteres gráficos e linha 510, que exibe a tela inicial. A linha 20 inicializa as variáveis que auxiliam a formatação da tela de jogo. A linha 30 exibe fundo da tela do jogo. A rotina das linhas 40 a 82 controla o jogo. Da 90 a 120 fazem a formatação da tela do jogo. A rotina entre as linhas 200 e 240 exibe tela final.

```
  10 PAPER 0: INK 7: BORDER 0: B
RIGHT 1: CLS : LET hi=0: GOSUB 1
050: GOSUB 510
  20 LET t=300: LET p=1: LET sc=
0: LET l=0: LET d=0: LET a$="
    ": LET b$=a$: LET c$="
  ": LET d$="    ": LET e$="



LET a=5: LET b=15: LET c=1
  30 BORDER 0: BRIGHT 1: PRINT A
T 0,0; INK 0; PAPER 6;"  PLACAR
      RECORD          ";AT 0,10;
sc;AT 0,25;hi;AT 21,0;"   LAGO 1
       TEMPO 300        ": FOR f
=1 TO 15: PRINT AT f,0; PAPER 5;
"
 ": NEXT f: PRINT PAPER 1;,,,,,,
,,,,
  40 LET e$=e$(32)+e$(1 TO 31)+e
$(34 TO 64)+e$(33): PRINT AT 16,
0; PAPER 1; INK 6; BRIGHT 1;e$:
IF d=1 THEN GOTO 90
  45 PRINT AT a,b; PAPER 5; BRIG
HT 1;"  ";AT a+1,b;"  ": IF INKE
Y$="8" THEN LET b$=b$(5 TO )+b$(
1 TO 4): LET a$=b$: LET c=1: LET
 d$="    ": LET b=b+1-(30 AND b=
30): GOTO 70
  50 IF INKEY$="5" THEN LET c$=c
$(5 TO )+c$(1 TO 4): LET a$=c$:
LET b=b-1+(30 AND b=0): LET c=-1
: LET d$="    ": GOTO 70
  60 LET a=a+(INKEY$="6" AND a<9
)-(INKEY$="7" AND a>1): IF INKEY
$="9" THEN LET d=1
  70 LET a$=a$(5 TO )+a$(1 TO 4)
: PRINT AT a,b; INK 0; PAPER 5;a
$(1 TO 2);AT a+1,b;a$(3 TO 4)
  80 PRINT AT 18,0; PAPER 1;"
                              ";AT
 0,10; PAPER 6; INK 0;sc: IF sc>
0 AND sc/7=INT (sc/7) AND l=0 TH
EN LET e$="


 ": LET p=p+1: LET t=3
00-(((p-1)*20) AND p<10)-(160 AN
D p>9): PRINT AT 21,9; PAPER 6;
INK 0;p;AT 21,21;t: LET l=1
  81 LET t=t-1: PRINT AT 21,21;
PAPER 6; INK 0;t;"  ": IF t>0 THE
N GOTO 40
  82 GOTO 200
  90 SOUND .001,60-a: LET p1=5:
LET p2=5: PRINT AT a,b; PAPER p1
; BRIGHT 1;"  ";AT a+1,b; PAPER
p2;"  ": LET a=a+1: LET b=b+c+(2
6 AND b<=2)-(26 AND b>=29): PRIN
T AT a,b; INK 0; PAPER p1; BRIGH
T 1;d$(1 TO 2);AT a+1,b; PAPER p
2;d$(3 TO 4): IF a=16 THEN LET p
1=5: LET p2=1: IF e$(b TO b+1)<>
"  " THEN LET e$(b-1 TO b+2)="
  ": PRINT AT a,b; PAPER 1; BRIG
HT 1;"  ";AT a+1,b;"  ": LET d=0:
 LET a=8: LET sc=sc+1: GOTO 81
  95 IF a=15 THEN SOUND .01,10:
SOUND .01,12
 100 IF a=17 THEN LET p1=1: LET
p2=1: IF e$(b+32)<>" " THEN LET
e$(b+31 TO b+33)="   ": PRINT AT
 a,b-1; PAPER 1; BRIGHT 1;"    "
;AT a+1,b;"  ": LET d=0: LET a=8
: LET sc=sc+1: LET l=0: GOTO 81
 110 IF a>17 THEN PRINT AT a,b-1
; PAPER 1; BRIGHT 1;"    ";AT a+
1,b-1;"    " LET a=8: LET d=0
 120 GOTO 81
 200 FOR g=0 TO 10: FOR f=0 TO 7
: SOUND .001,f+40: OUT '254,f*8:
OUT 254,200: PRINT AT a,b; INK f
; PAPER 5; OVER 1;"  ";AT a+1,b;
"  ": NEXT f: NEXT g: CLS
 210 PAPER 0: INK 7: BORDER 0: C
```

PROGRAMAS

```
LS : PRINT AT 10,0; INK 5;"

"
213 PRINT AT 13,0;"
"
 215 PRINT AT 14,0;"
"
 217 PRINT AT 15,0;"

"
 220 IF sc>hi THEN SOUND .5,10:
SOUND .5,0: SOUND .2,12: SOUND .
2,12: SOUND 1,20: PRINT AT 15,0;
 INK 0; PAPER 6; BRIGHT 1; FLASH
 1;"        UM NOVO RECORD
  ": LET hi=sc
 230 PRINT AT 13,0; INK 7; BRIGH
T 1;" PRECIONE TECLA.P/ CONTINUA
R  "
 240 PAUSE 1: PAUSE 0: GOTO 20
 510 PRINT AT 5,4;"Voce e um pel
icano  pescadore adora comer pei
xes. Porem vocetera um determina
do  tempo  paraisso. No primeiro
 lago voce teraque comer todos o
s  peixes  parapoder ir ao proxi
mo.Porem a cadanovo lago voce te
ra menos tempo."
 520 PRINT AT 16,4; INK 6;"USE A
S SEGUINTES TECLAS:        5-ESQ
UERDA      8-DIREITA     6-DESCE
     7-SOBE     9-PESCA"
 530 PRINT AT 20,0; INK 0; PAPER
 6; FLASH 1;"   PRECIONE TECLA P
ARA COMECAR   "
 540 LET a$="      ": LET b$="
     "
 550 PRINT AT 1,5; INK 6;a$(1 TO
 2);AT 1,25;b$(1 TO 2);AT 2,5;a$
(3 TO 4);AT 2,25;b$(3 TO 4): OUT
 254,24: IF INKEY$="" THEN LET a
$=a$(5 TO )+a$(1 TO 4): LET b$=b
$(5 TO )+b$(1 TO 4): OUT 254,200
: GOTO 550
 560 RETURN
 600 STOP
1000 DATA 0,0,1,3,187,255,235,85
,43,54,44,52,40,88,112,224,224,1
76,89,41,181,215,235,127,0,192,2
24,56,191,248,224,192
1010 DATA 0,0,128,192,221,255,21
5,170,212,108,52,44,20,24,14,7,7
,13,154,148,173,235,215,254,0,3,
7,28,253,31,7,3
1020 DATA 0,32,65,115,127,115,65
,128,0,96,248,204,232,252,248,96
,0,6,31,51,23,63,31,6,0,4,130,20
6,254,206,130,1
1030 DATA 152,148,218,213,250,12
5,122,61,0,0,0,0,128,128,128,192
,30,7,0,0,0,0,0,0,240,216,232,24
8,124,12,2,1
1040 DATA 0,0,0,0,1,1,1,3,25,41,
91,171,95,190,94,188,15,27,23,31
,62,48,64,128,120,224,0,0,0,0,0,
0
1050 FOR f=USR "a" TO USR "t"+7:
 READ a: POKE f,a: NEXT f: RETUR
N
```

```
ABCDEFGHIJKLMNOPQRSTU
                    U
A=
```

```
Voce e um pelicano  pescador
e adora comer peixes. Porem voce
tera um determinado  tempo  para
isso. No primeiro lago voce tera
que comer todos os  peixes  para
poder ir ao proximo.Porem a cada
novo lago voce tera menos tempo.

  USE AS SEGUINTES TECLAS:
  5-ESQUERDA       8-DIREITA
6-DESCE     7-SOBE      9-PESCA

  PRECIONE TECLA PARA COMECAR
```

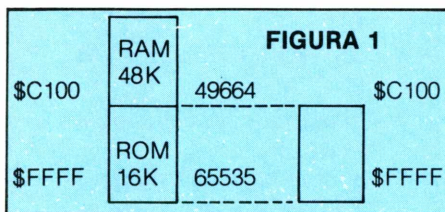
TELA INICIAL

Tente Estas!

TK-90X
POKE 23609,N onde N é a tonalidade do som do teclado
TK-90X
POKE 23613,0 protege programa contra listagem
TK-90X
POKE 23562,N onde N é a pausa na repetição
TK-90X
POKE 23658,8 letras maiúsculas
TK-90X
POKE 23692,255 roda o programa sem SCROLL
TK-90X
POKE 23756,0 numera primeira linha igual a 0

Utilizando os 64K de RAM do seu TK-2000

Aroldo Possuelo de Carvalho

Antes de falar e praticarmos o uso da expansão do TK-2000, vamos dar uma olhada em seu mapa de memória. Vejam a Figura 1.

À esquerda, temos o mapa de memória do TK-2000, com 48K de memória RAM (Random Access Memory) e 16K de memória ROM (Read Only Memory).

Na RAM temos variáveis do BASIC, duas páginas de vídeo e memória para o usuário programar em BASIC ou em Linguagem de Máquina; na ROM temos o programa monitor e todas as rotinas de cálculos de Ponto-Flutuante, bem como as rotinas BASIC.

Mas, alguma vez, você já deve ter se perguntado:

1) como meu TK tem 64K de RAM, sendo que só estão disponíveis 48K?
2) como pode o 6502 endereçar mais que 64K de memória, sendo que o seu "bus de endereço" é de 16 bits, e isto só permite endereçar 64K de memória?
3) onde estão os outros 16K de RAM?
4) se existem, como usá-las?

Se você fez algumas destas perguntas, aqui estão as respostas:

Ao ler o "Manual Técnico do TK-2000", você deve ter visto que nos endereços de $C000 a $C0FF temos 256 bytes para I/O, ou seja, para controlarmos os periféricos de entrada/saída, como:

- Teclado ($C000)
- Página de Vídeo 1 e 2 ($C054, $C055)
- I/O para gravador K7 ($C020), ($C010)
- Vídeo colorido e P/B ($C051), ($C050)
- Seleção de RAM ($C05B)
- Seleção de ROM ($C05A)

Estes dois últimos I/O's mencionados são os responsáveis pela seleção dos 16 Kbytes de RAM que completam os 64K de RAM.

Bem, até aqui talvez alguns dos leitores não tenham nenhuma dúvida, mas agora vamos ver como é que tudo isto funciona.

Você pode ter notado que os 16K de RAM estão ocupando os mesmos endereços que os 16K da ROM.

Portanto, se ativarmos a RAM (expansão) diretamente, perderemos o controle do TK, pois a ROM é acessada constantemente.

Como exemplo, deve-se fazer o seguinte:

- ligue o TK-2000. Se você estiver usando disk-drive dê um RESET no teclado;
- digite (LM) para entrar no modo monitor;
- digite $C05B RETURN.

Daí para frente nada mais acontecerá, ou então coisas estranhas aparecerão na tela, pois o TK-2000 passou o comando para a RAM e lá não havia nenhuma rotina que pudesse continuar monitorando o teclado e o vídeo pelo menos.

Há necessidade de se usar uma pequena rotina em Linguagem de Máquina, que controle a ROM e a RAM sem problemas.

Em nosso programa BASIC temos duas linhas DATA que contêm a rotina (em Linguagem de Máquina), que controlam a expansão e lhe mostrará como fazer para usar a expansão em seus programas.

Gostaria de destacar que você poderá usar esta expansão de memória para salvar espaços da memória RAM normal, ou até mesmo colocar nela rotinas de Linguagem de Máquina que torne o seu TK-2000 mais versátil.

Abrindo um pequeno parêntese, é assim que o Apple II trabalha. Um programa que está na RAM de 48K carrega outras linguagens na expansão (mais conhecida como "Language Card") como por exemplo, FORTH, Integer BASIC, COBOL, e outras rotinas. Isto é muito interessante, pois você poderá fazer o mesmo no TK-2000.

Bem, para isto é preciso aprender ***como*** fazer.

Mostrarei como salvar e recuperar dados na expansão, e isto será o bastante para que você possa criar novas rotinas e novas aplicações para o TK-2000.

Muitas pessoas não acreditam no TK-2000, pois pensam que o mesmo é compatível com o Apple II quando, na verdade, somente o BASIC Applesoft é compatível, e não em 100 por cento.

O TK-2000 possui características bem diferentes e é um computador com muito potencial a ser explorado, sendo que só uma minoria tem feito isto.

Bem, voltando ao nosso problema, vamos analisar o nosso programa. Ele está escrito em BASIC, mas o mesmo carrega as linhas DATA a partir da posição $300 (768), que é a nossa rotina que controla RAM/ROM e permite salvar dados RAM — Expansão e ler os dados da Expansão — RAM.

Para isto, separei uma área de memória que será o nosso ***buffer*** de dados. Esta posição é de $4000 a $7DFF (16384 a 32511); isto equivale a 16127 bytes de memória.

Ao ser rodado (RUN), o programa BASIC lerá os dados contidos nas linhas DATA e fará POKE's na posição $300 (768). Depois, apresentará um breve texto de instruções para uso do mesmo.

Vamos agora analisar a rotina em Linguagem de Máquina.

Se você nunca trabalhou em Linguagem de Máquina não se preocupe, pois a rotina é muito simples, e está bem detalhada.

Temos abaixo a listagem do programa em Assembler que inicia no endereço $300 (768) e termina em $353 (851); este bloco inclui duas rotinas. A primeira se encarrega de transferir os dados da memória RAM normal (MEMO) para a memória RAM expansão (EXP). E a segunda faz a transformação em sentido oposto, ou seja, EXP - MEMO.

Nas linhas de $300 a $30E, a rotina prepara quatro posições da página zero que são as seguintes:

$00 e $01: posições LOW e HI, que carregarão os dados de MEMO.

$02 e $03: posições LOW e HI, que armazenarão os dados na EXP.

Na linha $310, o ***Index Register*** Y é zerado, onde Y será o contador de 0 a 255.

A linha $312 carrega o acumulador apontado por $00 e $01, mais Y.

A linha $314 habilita os 16K de RAM EXPANSÃO.

Na linha $317, o valor contido no acumulador será armazenado no endereço de memória apontado por $02 e $03, mais Y.

A linha $319 habilita a memória ROM.

A linha $31C incrementa Y (Y + 1).

A linha $31D salta para a linha $312 se Y for diferente de zero, isto ocorrerá 255 vezes.

Na linha $31F, se Y for igual a zero, então a posição $01 é incrementada.

Na linha $321, a posição de memória $03 é incrementada.

A linha $323 carrega o acumulador com o conteúdo da posição $01.

A linha $325 compara o acumulador com # $61.

A linha $327 retorna à linha $312 se, na comparação, os valores forem diferentes.

A linha $329 retorna ao BASIC, se na comparação, os valores forem iguais.

Da linha $32A à linha $353, a rotina é similar à anterior.

O programa BASIC também se encarrega de comparar valores nas faixas de $2000 a $9600 e considerando 16K de comprimento; ele contém também as principais instruções.

PROGRAMAS

Em seus programas, você só precisará usar a rotina em Linguagem de Máquina.

Quero lembrar que um número precedido de "$" (dólar) significa um número em hexadecimal.

Após ter digitado corretamente o programa BASIC, digite LM seguido de um RETURN.

$4000: 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00

Você zera 20 bytes, de $4000 a $4016.

Digite CONTROL - C para voltar ao BASIC e RUN, selecione a opção 2 do MENU, retorne ao monitor com RESET e LM. Digite $4000 L RETURN, você verá que os valores foram alterados, pois a expansão continha estes valores.

Agora, rode o programa novamente RUN (RETURN) e selecione a opção 3. Mude os endereços de transferência para:

INÍCIO: 8192

FINAL: 16383, que correspondem à primeira página de vídeo.

Aparecerão barras ou pontos, enfim, você visualizará os dados continuados nos primeiros 8K da expansão e, logo após terminada a transferência, o programa BASIC voltará às instruções iniciais.

Daí em diante, você pode transferir rotinas, telas de vídeo ou tabelas de dados, da memória para a expansão e vice-versa.

O programa roda bem com disk-drive, e funcionou num TK-2000, com 64K de RAM.

LISTAGEM

```
0300-   A9 00      LDA   #$00
0302-   85 00      STA   $00
0304-   A9 00      LDA   #$00
0306-   85 02      STA   $02
0308-   A9 20      LDA   #$20
030A-   85 01      STA   $01
030C-   A9 C2      LDA   #$C2
030E-   85 03      STA   $03
0310-   A0 00      LDY   #$00
0312-   B1 00      LDA   ($00),Y
0314-   AE 5B C0   LDX   $C05B
0317-   91 02      STA   ($02),Y
0319-   AE 5A C0   LDX   $C05A
031C-   C8         INY
031D-   D0 F3      BNE   $0312
031F-   E6 01      INC   $01
0321-   E6 03      INC   $03
0323-   A5 01      LDA   $01
0325-   C9 40      CMP   #$40
0327-   D0 E9      BNE   $0312
0329-   60         RTS
032A-   A9 00      LDA   #$00
032C-   85 00      STA   $00
032E-   A9 00      LDA   #$00
0330-   85 02      STA   $02
0332-   A9 20      LDA   #$20
0334-   85 01      STA   $01
0336-   A9 C2      LDA   #$C2
0338-   85 03      STA   $03
033A-   A0 00      LDY   #$00
033C-   AE 5B C0   LDX   $C05B
033F-   B1 02      LDA   ($02),Y
0341-   AE 5A C0   LDX   $C05A
0344-   91 00      STA   ($00),Y
0346-   C8         INY
0347-   D0 F3      BNE   $033C
0349-   E6 01      INC   $01
034B-   E6 03      INC   $03
034D-   A5 03      LDA   $03
034F-   C9 E3      CMP   #$E3
0351-   D0 E9      BNE   $033C
0353-   60         RTS
```

```
0 REM *****************
************
1 REM * USANDO A EXPANS
AO DE 16K *
2 REM * AROLDO POSSUELO
DE CARVALHO*
3 REM *
           *
4 REM *          MAIO 19
86          *
5 REM *****************
************
7 HOME
10 DIM A(83)
15 FOR I = 768 TO 851:N
= (I - 768): READ A(N): PO
KE I,A(N): NEXT I
20 INVERSE : PRINT "  TR
ANSFERENCIA D
A MEMORIA / EXPANSAO  ":
NORMAL
25 PRINT : PRINT "   ES
TE PROGRAMA T
RANFERE O CONTUDO DA MEMO
RIA RAM
PARA A EXPANSAO DE 16K DO
  TK200
0 , E VICE-VERSA."
30 PRINT : PRINT "    OS
 DADOS A SERE
M TRANFERIDOS SERAO  OS D
E $4000
- $7DFF -> $C200'- $FFFF.
  OU VI
CE-VERSA , ISTO LHE PERMI
TIRA'
  ULTILIZAR MELHOR A MEMO
RIA DO T
K2000.
35 PRINT : PRINT "    AQ
UI ESTA' A EQ
UIVALENCIA DOS END.  DE H
EXADECIM
AL PARA DECIMAL REPECTIVA
S.": PRINT : PRINT "$4000
-> 16384": PRINT "$7DF
F -> 32511": PRINT "$C200
 -> 4966
4": PRINT "$FFFF -> 65535
"
40 PRINT : PRINT : PRINT
 "          P
RESSIONE UMA TECLA
   ": SOUND 50,20: GET A$:
 HOME
45 HTAB 1: VTAB 5: PRINT
 "1 - MEMORIA
 PARA EXPANSAO": HTAB 1:
VTAB 7: PRINT "2 - EXPANSA
O PARA MEMORIA": HTAB
1: VTAB 9: PRINT "3 - MUD
AR END.
DOS DADOS TRANSFERIDOS "
50 VTAB 13: INPUT "QUAL
E' A SUA ESCO
LHA ?:";A$
55 IF A$ = "1" THEN 100
60 IF A$ = "2" THEN 150
63 IF A$ = "3" THEN 200
65 GOTO 50
70 DATA  169,0,133, 0,1
69,0,133, 2,1
69,64,133, 1,169,194,133,
 3,160,0
,177, 0,174,91,192,145, 2
,174,90,
192,200,208,243,230,1,230
, 3,165,
```

▷

```
1,201,97,208,233,96,169,0
,133,0,1
69,0
80  DATA  133, 2,169,64,1
33, 1,169,194
,133, 3,160,0,174,91,192,
177, 2,1
74,90,192,145, 0,200,208,
243,230,
1,230, 3,165,3,201,210,20
8,233,96

100  REM  MEMO -> EXP.
105  HOME : VTAB 11: HTAB
 1: PRINT "TR
ANSFERINDO DADOS PARA EXP
ANSAO...
    "
120  CALL 768: GOSUB 300:
 GOSUB 165
130  GOSUB 300: HOME : GO
TO 20
150  REM  EXP -> MEMO
155  HOME : VTAB 11: HTAB
 1: PRINT "TR
ANSFERINDO DADOS DA EXPAN
SAO...
    "
160  CALL 810: GOSUB 300:
 GOSUB 165: HOME : GOTO 20

165  SOUND 50,20: VTAB 11
: HTAB 1: PRINT "TRANSFERE
NCIA TERMINADA !

    ": RETURN
200  REM  MUDANDO O END.
```

```
205  HOME : PRINT : PRINT

210  PRINT "    DIGITE O
NOVO ENDERECO
 DO BLOCO DE  DADOS A SER
EM TRANS
FERIDOS."
220  HTAB 1: VTAB 7: PRIN
T "USE ENDERE
COS DE 8192 ($2000) A 384
00   ($9
600)": PRINT : PRINT
225  INPUT "DIGITE NUM. D
ECIMAL DE INI
CIO: ";N$
227  GOSUB 500: POKE 769,
L: POKE 777,H
: POKE 811,L: POKE 819,H
280  PRINT : INPUT "DIGIT
E NUM. DECIMA
L FINAL: ";N$
282  GOSUB 500: POKE 806,
H + 1
290 A =  PEEK (806) -  PE
EK (777)
291  IF A * 256 + 255 >
= 15871 THEN
380
295  POKE 848, PEEK (823)
 + A + 1
299  PRINT : INVERSE : PR
INT "ENDERECO
S MUDADOS...": NORMAL : S
OUND 50,
10: GOSUB 300: HOME : GOT
O 20
```

```
300  FOR I = 0 TO 600: NE
XT : RETURN
350  HTAB 1: VTAB 23: INV
ERSE : PRINT
"NUMERO FORA DA FAIXA DE
8192 A 3
8400    ": SOUND 40,10: NO
RMAL : GOSUB 300: GOSUB 37
0: GOTO 220
360  HTAB 1: VTAB 23: INV
ERSE : PRINT
"ERRO. DIGITE.OS 5 ALGARI
SMOS
      ": SOUND 40,10: NO
RMAL : GOSUB 300: GOSUB 37
0: GOTO 220
370  HTAB 1: VTAB 23: PRI
NT "

    ": RETURN
380  HTAB 1: VTAB 23: INV
ERSE : PRINT
"COMPRIMENTO DOS ENDERECO
S MAIOR
QUE 16K": NORMAL : GOSUB
300: GOSUB 370: GOTO 220
500  REM  CONV. DEC->HEX
505  IF  LEN (N$) <  = 0
THEN 360
520  IF  VAL (N$) >  = 38
401 OR  VAL (
N$) <  = 8191 THEN 350
530 H =  INT ( VAL (N$) /
 256):L =  VAL (N$) - H *
256
555  RETURN
```

◇

PROGRAMAS

Conversão numérica

Agora é fácil fazer conversão entre sistemas de numeração com o uso do TK 2000.

Autora: *Zilah M. Miazaki*
Adaptação: *Elaine Kelly Lima*

Ao ver digitar, ou mesmo fazer um programa em Assembler, você deve ter notado que os números ali representados normalmente são compostos por caracteres que variam entre 0 e F. Porém, para que utilizarmos caracteres alfabéticos em um número, se o que aprendemos desde a infância é representá-los com dígitos variantes entre 0 e 9?

Simplesmente, podemos dizer que isso ocorre devido a existência de vários sistemas de numeração que se adaptam cada um a uma utilização específica.

Nós sempre usamos o sistema de numeração de base dez, isto porque inicialmente a representação das quantias e das contagens eram feitas usando os dedos. Como todos os seres humanos temos dez dedos, e por isso convencionou-se um sistema com dez caracteres.

Quando foram projetadas as primeiras máquinas calculadoras tentaram fazer com que seu processamento fosse com números de base dez. Porém, com a evolução eletrônica, verificou-se ser mais simples e rápido representar, internamente no equipamento, apenas dois estados: ligado ou desligado. Os números para representação destes estados são respectivamente 0 e 1, sendo que o sistema é binário.

Devido à quantidade de memória ocupada pelos números binários e pela complexidade para se programar corretamente com apenas dois caracteres, convencionou-se para as linguagens de baixo-nível a utilização do sistema hexadecimal, ou seja, sistema com base 16 cujo caractere menos significativo é o 0 e o mais significativo é o F.

A conversão entre sistemas numéricos é uma tarefa fácil, mas muitas vezes se torna cansativa. Como todos sabemos, existem vários sistemas numéricos e entre os mais conhecidos estão:

- binários, cuja base é 2 e aceita os caracteres 0 e 1;
- octal, cuja base é 8 e seus números são formados com caracteres entre 0 e 7;
- decimal, que é o utilizado normalmente por nós;
- hexadecimal, com base 16 e faz a representação numérica com caracteres entre 0 e F.

O programa apresentado aqui é para o TK 2000, e foi desenvolvido para usuários que sentem a necessidade de se aprofundar no BASIC do TK ou que estão iniciando o aprendizado em Linguagem de Máquina. Porém, também é muito útil para programadores de Assembler fazerem suas conversões.

Sua estrutura é simples, flexível e corrente para que você possa, dessa maneira, entender sua lógica.

Ao rodá-lo, a tela inicial fará uma apresentação e a seguir perguntará a base do número a ser convertida, e exibirá o respectivo resultado. A cada pergunta, o programa faz uma verificação e observa se nenhuma informação absurda foi digitada. A seguir, o resultado é rapidamente apresentado no vídeo e se o usuário quiser prosseguir deverá teclar "S".

Esquema lógico do programa

Se você observar as linhas entre 10 e 40 notará que elas são responsáveis pela apresentação de dois delimitadores: de início e final da tela.

Na linha 50 o comando POKE 34,5 e POKE 35,19 abrem janela horizontal e na linha 55 o POKE 32,5 e o POKE 33,35 abrem janela vertical. O CALL-936 apaga a tela.

A linha a seguir é a 80 e ela irá fazer a verificação sobre a finalização ou não do programa. Na linha 86 você poderá observar a definição da variável que tem como conteúdo os caracteres que podem formar os números de todos os sistemas.

A rotina entre as linhas 90 e 200 executa a entrada e a consistência dos dados necessários à conversão, e a rotina entre 210 e 320 faz a conversão do número.

A linha 330 faz a exibição do resultado e a 340 provoca retorno ao início.

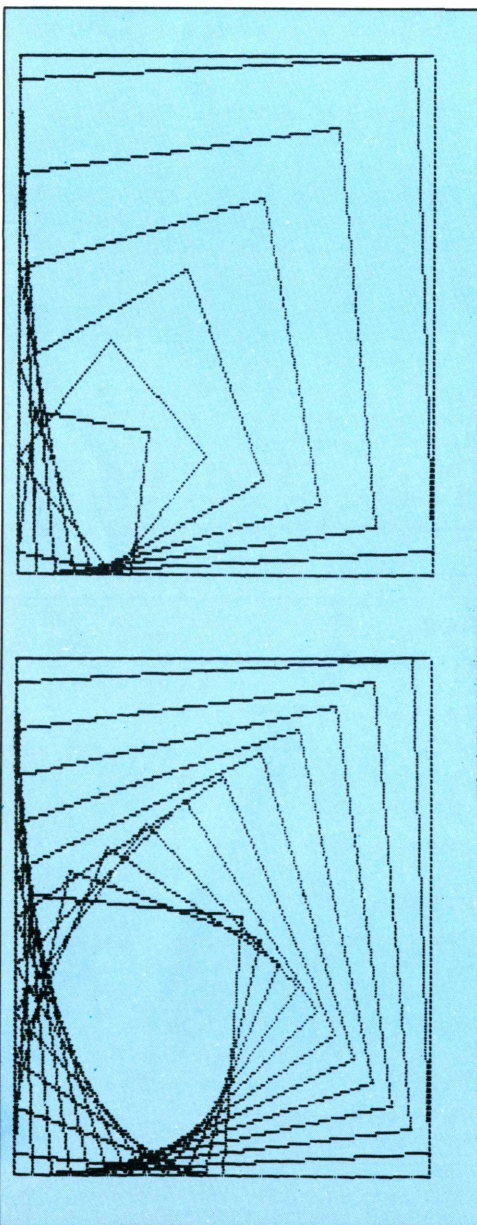
Conversão numérica no TK 3000

Para converter este programa para o TK 3000, ou qualquer compatível com os equipamentos da linha Apple, você deverá excluir as linhas entre 10 e 40 inclusive.

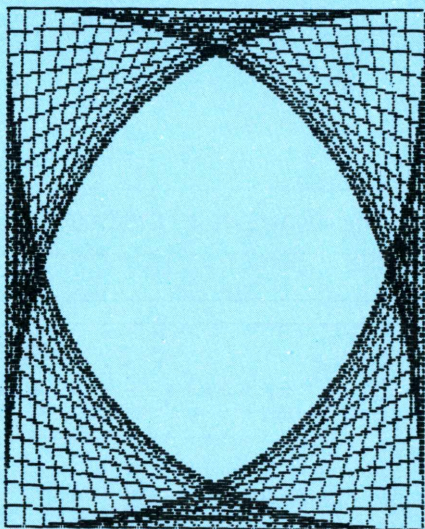
A retirada desta rotina ocorre pelo fato dos equipamentos compatíveis com o Apple apresentarem a tela de texto separada da tela de alta-resolução.

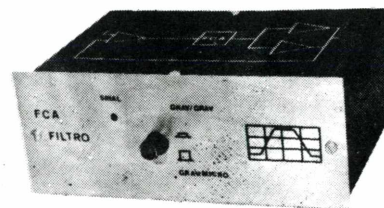
```
10  HGR2 : HCOLOR  = 3: F
OR N = 0 TO 1
91
20  IF N = 10 THEN  LET N
 = 180
30  HPLOT 0,N TO 279,N
40  NEXT N
50  POKE 34,5: POKE 35,19

55  POKE 32,5: POKE 33,35

60  CALL  - 936
70  HTAB 14: VTAB 8: PRIN
T "CONVERTEND
O"; SPC( 51);"NUMEROS"; S
PC( 54);
"PARA"; SPC( 56);"BASES";
 SPC( 52
);"DIFERENTES"
80  INVERSE : HTAB 6: VTA
B 18: PRINT "
PRECIONE  S  PARA CONTINU
AR": NORMAL : GET B$: IF B
$ <  > "S" THEN  POKE
32,0: POKE 33,40: POKE 34
,0: POKE
35,24: END
85  CLEAR : CALL  - 936
86 T$ = "0123456789ABCDEF
"
90  INVERSE : HTAB 6: VTA
B 6: PRINT "D
IGITE A BASE A CONVERTER"
: HTAB 6
: VTAB 8: PRINT "DE 2 A 1
6": NORMAL
100  HTAB 20: VTAB 8: INP
UT E: IF E <
2 OR E > 16 THEN 100
110  HTAB 6: VTAB 10: PRI
NT "DIGITE NU
MERO": HTAB 20: VTAB 10:
INPUT N$

120 L =  LEN (N$)
130  FOR F = 1 TO L:F$ =
 MID$ (N$,F,1
)
140  IF F$ < "0" OR F$ >
 MID$ (T$,E,1
) OR L < 1 OR L > 30 THEN
```

▷

```
 F = 0:N
 = 0: GOTO 110
150  IF  ASC (F$) > 57 TH
EN M =  ASC (
F$) - 55: GOTO 170
160 M =  VAL (F$)
170 N = N + M * E ^ (L -
F): IF N > 10
 ^ 9 THEN N = 0:F = 0: GO
TO 110
180  NEXT F
190  HTAB 6: VTAB 12: INV
ERSE : PRINT
"DIGITE BASE A SER CONVER
TIDO": HTAB 6: VTAB 14: PR
INT "DE 2 A 16": NORMAL
200  HTAB 20: VTAB 14: IN
PUT E: IF E <
2 OR E > 16 THEN 200
210  FOR F = 0 TO 30: IF
E ^ (F + 1) >
N THEN 230
220  NEXT F
230  FOR C = F TO 0 STEP
 - 1
240  FOR B = E - 1 TO 0 S
TEP  - 1
250 M = E ^ C * B: IF M <
 = N THEN M$
 = M$ +  MID$ (T$,B + 1,1
):N = N - M:B = 0: GOTO 30
0
260  NEXT B
300 M =  INT (N):L = M -
N + 1: IF L <
.5 THEN N = M + 1
310  IF L > = .5 THEN N
= M
320  NEXT C
330  HTAB 6: VTAB 16: PRI
NT "O RESULTA
DO :": HTAB 20: VTAB 16:
PRINT M$
340  GOTO 80
```

Tente Esta!

Elaine Kelly Lima

Tente este no seu **TK-90X.** Modificando o **STEP** da linha 80, você poderá diminuir ou aumentar as dimensões dos quadrados.

```
 10 FOR n=0 TO 174
 20 PLOT n,0: PLOT n,175
 30 NEXT n
 40 FOR n=0 TO 174
 50 PLOT 0,n: PLOT 175,n
 60 NEXT n
 70 LET a=0
 80 FOR n=167 TO 0 STEP-8
 90 LET a=a+8
100 PLOT 0,n  DRAW a,-n
110 DRAW n,a
120 DRAW -a,n
130 DRAW -n,-a
140 NEXT n
```

VICE-VERSA

V.V. 013

Lay-out da tela de vídeo

Elaine Kelly Lima

A projeção de um *lay-out* de vídeo perfeito é fundamental em qualquer espécie de programa, isto porque normalmente é este terminal que permite a comunicação entre o programa e o usuário.

O planejamento das telas de jogos é caracterizado pela exploração dos recursos gráficos do equipamento. Já os sistemas comerciais e os utilitários necessitam de telas que permitam ao operador visualizar todos os dados e mensagens.

A fim de apresentar aos usuários programas com boas telas, muitos profissionais optaram por trabalhar com o BASIC, pois todas as suas versões são muito flexíveis e oferecem diversos recursos. Entre seus comandos, o INPUT (entrada de dados via teclado) e o PRINT (exibição de dados) são os que mais afetam a formatação de tela de texto e por este motivo, tudo o que foi selecionado para este artigo está relacionado a um destes dois comandos.

Dimensões diferentes

Ao adaptarmos telas devemos considerar que a dimensão do vídeo dos equipamentos pode ser diferente, tornando necessária a modificação do *lay-out.* Observe as figuras 1 e 2, ambas apresentam uma tela de menu, porém a da primeira aceita até 40 colunas por linha, já a segunda aceita apenas 32.

Para adaptar *lay-outs* você precisará conhecer a dimensão das telas dos diferentes equipamentos e os comandos que trabalham com o posicionamento de informações.

Versão do BASIC	Dimensões	
	Linhas	Colunas
Sinclair	24	32
Spectrum	24	32
Apple	24	40
TRS-80	16	64
Microsoft (com cartão 80 colunas)	24	80
Microsoft (sem cartão 80 colunas)	24	dimensão de colunas normais do equipamento usado.

Alguns equipamentos utilizam as duas últimas linhas para mensagens do sistema.

Os posicionadores

Ao indicar a posição de exibição de mensagens ou variáveis, as instruções utilizadas pelas diversas versões são diferentes. A seguir, mostramos os posicionadores mais comuns, onde e quando são usados.

Obs.: Nos formatos que são apresentados a seguir, ***L*** é o número de linha, ***C*** é o número da coluna e ***P*** é a posição.

PRINT AT - Exibe variável ou string na linha e coluna especificada pelo argumento da função AT.

FORMATO - PRINT AT ***L, C;*** "variável ou constante"

VERSÃO: Sinclair e Spectrum

PRINT @ - Exibe variável ou string na posição indicada pelo argumento, que segue o símbolo de arroba. Este argumento pode variar de 0 a 1023.

FIGURA 1

```
****************************************
***        CONTROLE VETERINARIO      ***
***           MENU PRINCIPAL         ***
****************************************




             1.INCLUSAO
             2.ALTERACAO
             3.EXCLUSAO
             4.CONTROLE
             5.FIM




----------------------------------------
MENSAGENS: OPCAO DESEJADA -->
----------------------------------------
```

FIGURA 2

```
********************************
***     Menu Principal      ***
********************************



          1.Inclusao

          2.Alteracao

          3.Exclusao

          4.Relatorio

          5.FIM



--------------------------------
Obs.:
--------------------------------
```

▷

FORMATO: PRINT @ *P;* "variável ou constante"

VERSÃO: TRS-80

TAB - Posiciona o cursor na coluna especificada pelo argumento, sendo que este pode ser um número entre 0 e 255.

FORMATO: PRINT TAB *C;* "variável ou constante". Ou

TAB *C*

VERSÃO: todas

HTAB - Posiciona o cursor na coluna especificada pelo argumento.

FORMATO: HTAB *C*

VERSÃO: Apple

VTAB - Posiciona o cursor na linha especificada pelo argumento.

FORMATO: VTAB *L*

VERSÃO: Apple

SPC - O argumento indica a quantidade de colunas em que o cursor se movimentará, adiante da posição atual.

FORMATO: SPC *P*

VERSÃO: Apple e Microsoft

DEF FNT$(L,C) - Exibe variável ou constante na posição indicada pelo FNT$(*L,C*).

FORMATO:

10 DEF FNT$(L,C) = CHR$(27) + "Y" + CHR$(L + 31) + CHR$(C + 31)

20 PRINT FNT$*(L,C);* "variável ou constante"

Obs.: A linha 10 deve ser colocada no programa uma única vez, pois apenas define a fórmula para formatação de tela, e nela as letras L e C não são variáveis, e por isso a definição deve ser digitada tal qual sua apresentação na linha 10. Toda vez que se especificar posição, deve-se usar o formato da linha 20 e nela as letras *L* e *C* são variáveis e devem ser substituídas pelo número da linha e coluna respectivos.

VERSÃO: Microsoft

Entrada de dados via console

A leitura de dados do terminal é feita normalmente com a instrução INPUT, que provoca diferentes reações nos equipamentos. Por exemplo, no Sinclair e Spectrum esta instrução causa a entrada de dados sempre na linha 22. Nas outras versões do BASIC a entrada acontece na posição atual do cursor. Além disso, você já deve ter notado que esta instrução em qualquer equipamento faz com que um sinal de interrogação seja apresentado no local onde a informação for pedida. Por este motivo, algumas versões oferecem instruções especiais, para entrada de dados, que não afetam a formatação de tela.

LINE INPUT - Possibilita a entrada de string sem que a interrogação apareça na tela.

FORMATO: LINE INPUT A$

VERSÃO: Microsoft

INKEY$ - Verifica se alguma tecla foi pressionada e não altera nada na tela, nem faz com que o que foi digitado apareça no vídeo.

FORMATO: LET I$ = INKEY$

VERSÃO: Sinclair, Spectrum, TRS-80 e Microsoft.

GET - mesma função que o INKEY$.

FORMATO: GET I$

VERSÃO: Apple

Rotina para exibição

Além dessas mudanças são necessárias alterações no programa:

- ao invés de CLS, o BASIC Applesoft usa, para apagar a tela, a instrução HOME;
- as instruções com formato PRINT AT *L, C;* "expressão" devem ser mudadas para a seguinte estrutura: HTAB *C* : VTAB *L* : PRINT "expressão".

Note que os comandos para exibição deixam de ter um complemento e têm um acréscimo de duas instruções de posicionamento à sua frente, que têm a mesma função do AT.

Na versão Microsoft que é caracterizada por processar sob o sistema operacional CP/M, a formatação das expressões dependerá do uso ou não do cartão de 80 colunas. Por este motivo, não daremos as modificações de ***lay-out,*** mas apenas as de programa.

FIGURA 3

```
  10 REM Lay out - Tela de Video
  20 CLS
  30 PRINT AT 0,0;"*************
**********************";AT 1,8;"
Menu Principal";AT 1,29;"*******
**************************"
  40 PRINT AT 19,0;"-------------
-------------------Obs.:";AT 21
,0;"----------------------------
----"
  50 PRINT AT 6,10;"1.Inclusao"
  60 PRINT AT 8,10;"2.Alteracao"
  70 PRINT AT 10,10;"3.Exclusao"
  80 PRINT AT 12,10;"4.Relatorio
"
  90 PRINT AT 14,10;"5.FIM"
 100 STOP
```

A rotina da figura três apresenta uma tela formatada no vídeo do Sinclair ou Spectrum e nos servirá de exemplo perfeito na demonstração das mudanças necessárias para passarmos, para outras versões, o ***lay-out*** de vídeo da figura 2.

Na versão Apple são necessárias modificações na formatação das expressões, já que o programa foi feito para vídeos de 22 linhas e 32 colunas e o vídeo do Apple tem 40 colunas.

Deve-se proceder da seguinte maneira:

- na linha 30 do programa deve-se acrescentar oito asteriscos às strings formadas por estes caracteres e a coluna de exibição da expressão do segundo AT passará a ser 12 e a coluna do terceiro AT, passará a ser 38;
- na linha 40 do programa deve-se acrescentar oito "-" no início das expressões a serem impressas pelos dois posicionadores nela encontrados;
- da linha 50 a 90 do programa, as colunas de impressão devem ser mudadas para 15.

A versão ***linha 20*** dependerá do equipamento. Na maioria das vezes, a instrução para apagar a tela é PRINT CHR$(12).

Torna-se necessário acrescentar uma linha antes da 30 para fazermos a definição da função de posicionamento de exibição. O formato da expressão pode ser:

15 DEF FNT$(L,C) = CHR$(27) + "Y" + CHR$ (L + 31) + CHR$(C + 31)

Em seguida, deve-se alterar todos os PRINT AT ***L,C*** para PRINT FNT$(***L,C***).

Na versão TRS-80 existe a necessidade de uma profunda mudança em tudo, já que a dimensão de sua tela é completamente diferente. No programa, todos os PRINT AT são mudados para PRINT@, e ao invés do número da linha e da coluna, nós teríamos que indicar a posição de exibição.

▷

VICE-VERSA

Formatando variáveis para exibição

A versão Microsoft do BASIC oferece aos seus usuários uma instrução que permite formatar as variáveis para impressão. Este comando é o PRINT USING, que como o nome diz, imprime (PRINT) usando (USING) determinado formato.

Como as outras versões do BASIC não oferecem a seus usuários este comando nem nenhum semelhante, resolvemos desenvolver uma rotina para os compatíveis com o Sinclair e Spectrum que simula este comando, aliás simula três características de comando PRINT USING, trabalhando com números:

- permite que você defina a quantidade de casas decimais que o seu número terá;
- permite que o número tenha, a cada três dígitos, um ponto de separação;
- possibilita ao usuário definir se deseja ou não que os caracteres em branco, à esquerda, sejam preenchidos com asteriscos.

As únicas observações negativas que temos a fazer é que o número não pode ter uma parte inteira maior que oito dígitos e sua parte decimal pode ficar alguns centésimos distorcida, isso devido ao armazenamento da variável em memória e também pelos processamentos realizados quando o programa roda.

Caso você junte esta rotina a algum programa saiba que a variável ***n*** deve conter o número a ser exibido, a variável ***d*** deve conter a quantidade de casas decimais e as variáveis ***i*** e ***i1*** devem conter a quantidade de números que compõem a parte inteira do número.

```
 10 REM Simulando USING
 20 CLS : INVERSE 1: PRINT AT 0,7;"Simulando USING": INVERSE 0
 30 PRINT AT 3,0;"Numero:";
 40 INPUT n: IF n>99999999 THEN GOTO 40
 50 PRINT n;AT 5,0;"Casas decimais:";
 60 INPUT d
 70 PRINT d;AT 7,0;"Casas inteiras:";
 80 INPUT i: IF i>8 THEN GOTO 80
 90 LET i1=i: PRINT i;AT 9,0;"Asteriscos(s/n):";
 100 INPUT a$: IF a$<>"s" AND a$<>"S" AND a$<>"n" AND a$<>"N" THEN GOTO 100
 110 PRINT a$
 120 LET n$=STR$ INT n
 125 LET w$=""
 130 LET x=0
 140 LET l=LEN n$
 150 IF l<i THEN LET i1=l
 160 IF l>8 THEN LET n$=n$(l-8 TO l)
 170 FOR f=l TO l-i1+1 STEP -1
 180 LET x=x+1
 190 LET w$=n$(f)+w$
 200 IF f=1 THEN GOTO 220
 210 IF x=3 OR x=6 THEN LET w$="."+w$
 220 NEXT f
 230 IF d=0 THEN GOTO 262
 240 LET d1=n-INT n
 250 IF d1>0 THEN LET d1=d1*10↑d: LET d1=INT d1
 260 LET w$=w$+","+STR$ d1
 262 LET n$=w$
 265 IF a$<>"s" AND a$<>"S" THEN GOTO 330
 270 LET t=d+1+i
 280 LET a=LEN w$
 300 FOR f=a+1 TO t
 310 LET n$="*"+n$
 320 NEXT f
 330 PRINT AT 11,5;n$
 340 STOP
```

```
Simulando USING

Numero:999324.45

Casas decimais:2

Casas inteiras:8

Asteriscos(s/n):s

*999.324,44
```

Tente Esta!

Se você possui um APPLE ou compatível e deseja abrir um arquivo seqüêncial, antes de iniciar seu programa de cadastramento, então use esta rotina.

```
10 REM CRIANDO ARQUIVO
20 REM EM DISCO NO
30 REM BASIC DO APPLE
50 D$ = CHR$ (4)
60 INPUT "NOME DO ARQUIVO";A$
70 PRINT D$;"OPEN";A$
100 PRINT D$;"WRITE";A$
110 PRINT "PRIMEIRA FRASE DO ARQUIVO"
120 PRINT D$;"CLOSE";A$
130 END
```

Polígonos Estrelados

Aprenda a desenhar polígonos, estrelas, círculos, e muitos outros efeitos "estrelados", como os das figuras ao lado.

Cesar de Afonseca e Silva Neto
Wilson José Tucci

Como fazer?

O programa pergunta pelo raio da circunferência circunscrita ao polígono. Repare que, uma circunferência pode ser descrita como um polígono de muitos lados, e este se torna o algoritmo utilizado na sub-rotina 650-690, que desenha uma circunferência com muita rapidez. Na realidade, estamos desenhando um polígono de 60 lados.

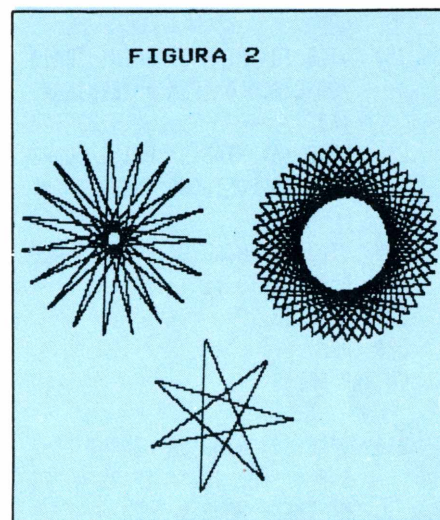
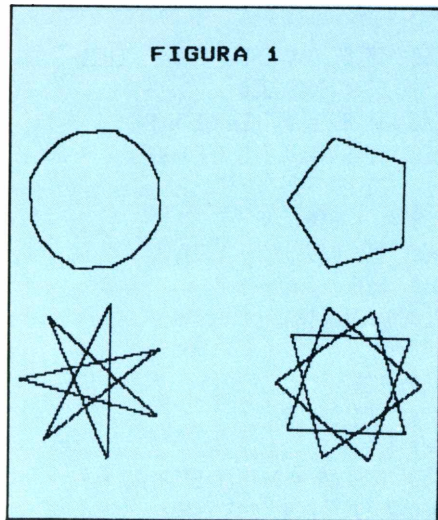
Em seguida, o programa pergunta pelo número de vértices do polígono e pelo salto. Bem, o número de vértices está claro o que significa, mas o valor salto, merece uma melhor explicação.

Quando entramos com o número de vértices do polígono, o programa montará duas matrizes: X( ) e Y( ). Estas matrizes conterão as coordenadas X e Y dos V vértices do polígono, respectivamente.

O valor do salto (variável ST) determinará em que ordem serão traçados os lados do polígono. Se o valor do salto é 1, a rotina de desenho irá unir todos os vértices do polígono, um a um. Se o valor de salto for maior, serão "pulados" ST-1 vértices. Através da combinação destes dois valores (V e ST), podemos obter os chamados "polígonos estrelados".

Observe atentamente a descrição das variáveis utilizadas no programa e até a próxima!

Variáveis utilizadas

- AC - Valor do ângulo central do polígono.
- AN - Variável utilizada como índice nos laços FOR...NEXT (ANgulo).
- P - Passo. Valor correspondente a 1 grau em radianos.
- PI - Contém o valor 3,1415926, utilizado nas equações de seno e coseno.
- P( ) - Matriz que contém a posição lógica de cada vértice. Se P(i) contém o valor 1 (verdadeiro) significa que o vértice já foi traçado.
- R - Raio da circunferência.
- ST - Valor do salto a ser dado.
- T - Período da circunferência (2 * PI).
- V - Número de vértices do polígono.
- XI - Coordenada X inicial.
- YI - Coordenada Y inicial.
- X( ) - Matriz que contém as coordenadas X de todos os V vértices.
- Y( ) - Matriz que contém as coordenadas Y de todos os V vértices.

Principais rotinas

120-160	Definição das variáveis de controle.
170-280	Formação da tela de texto. Esta tela continuará sempre na memória, bastando dar TEXT para acessá-la, instantaneamente.
230	Sub-rotina que imprime uma variável M$ de forma centralizada, em uma posição vertical da tela (V), utilizando S espaços entre cada caracter da mesma.
300-380	Entrada de alguns dados necessários.
450-490	Cálculo das coordenadas X e Y dos vértices, montando as matrizes X( ) e Y( ).
500	Chama rotina que desenha a circunferência.
510-570	Controle do traçado dos pontos.
590-600	Aguarda que uma tecla seja pressionada.
610-620	Truque ensinado na edição anterior, que apaga as matrizes que estão na memória sem termos que rodar o programa novamente, a fim de redimensioná-las.
650-690	Rotina que desenha a circunferência como sendo um polígono de 60 lados.

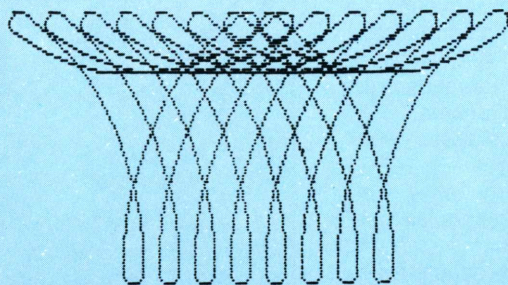
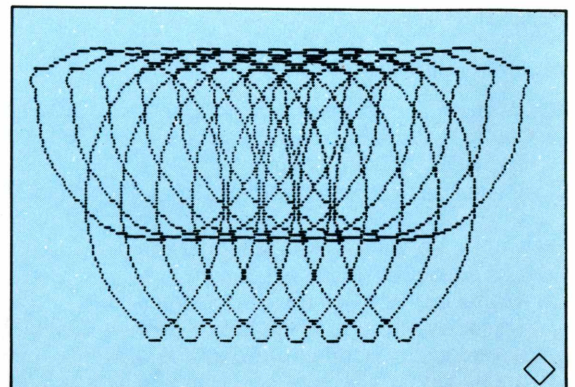
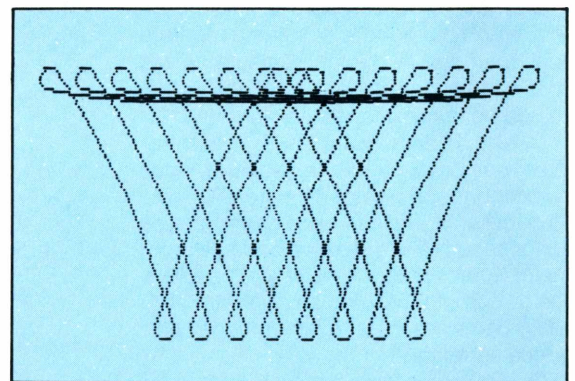
```
100  REM   POLIGONOS ESTRELADOS
110  REM  MICROHOBBY #33 / JULHO

120 PI = 4 *  ATN (1)
130 T = 2 * PI
140 P = PI / 180
150 R = 60
160 XI = 140:YI = 96
170  REM  TELA DE TEXTO
180  TEXT : HOME
190  INVERSE : PRINT ":"; SPC( 38
     );":": FOR I = 2 TO 22: VTAB
     I: HTAB 1: PRINT  SPC( 1);: HTAB
     40: PRINT  SPC( 1): NEXT : VTAB
     7: PRINT ":"; SPC( 38);":": VTAB
     23: PRINT ":"; SPC( 38);":";
     : NORMAL
200 M$ = " POLIGONOS ":V = 3:S =
     2: GOSUB 230
210 M$ = " ESTRELADOS ":V = 5:S =
     1: GOSUB 230
220  GOTO 240
230 H = (40 - ( LEN (M$) + ( LEN
     (M$) - 1) * S)) / 2: POKE 36
     ,H: VTAB V: FOR I = 1 TO  LEN
     (M$): INVERSE : PRINT  MID$
     (M$,I,1); SPC( S * ( LEN (M$
     ) (  ) I));: NEXT : NORMAL :
      RETURN
240  VTAB 9: HTAB 5: PRINT "ENTRE
      COM OS DADOS A SEGUIR"
```

▷

POR DENTRO DO APPLE

```
250  VTAB 13: HTAB 3: PRINT "RAIO
      DO CIRCULO (FIM = TERMINAR)
     :60"
260  VTAB 16: HTAB 3: PRINT "VERT
     ICES DO POLIGONO INSCRITO :"

270  VTAB 19: HTAB 3: PRINT "VALO
     R DO SALTO :1"
280  POKE 33,39
290  REM
ENTRAR DADOS

300  VTAB 13: HTAB 36: INPUT "";R
     $:R =  VAL (R$): IF R$ = "FI
     M" THEN  HOME : END
310  IF R > 280 - XI OR R > 192 -
     YI THEN  PRINT  CHR$ (7): GOTO
     300
320  IF R = 0 THEN R = 60
330  VTAB 16: HTAB 34: INPUT "";V
     $:V =  VAL (V$)
340  IF V = 0 THEN  PRINT  CHR$ (
     7): GOTO 330
350  VTAB 19: HTAB 19: INPUT "";S
     T$:ST =  VAL (ST$)
360  IF ST > V THEN  PRINT  CHR$
     (7): GOTO 350
370  IF ST = 0 THEN ST = 1
380  IF ST > V / 2 THEN ST = V -
     ST
390 AC = 360 / V
400  DIM P(V),X(V),Y(V)
410  REM
BLOCO PRINCIPAL

420  HGR2
430  HCOLOR= 3
440 I = 1
450  FOR AN = P TO T STEP AC * P
460 X(I) = R *  COS (AN)
470 Y(I) = R *  SIN (AN)
480 I = I + 1
490  NEXT
500  GOSUB 650
510 I = 1
520  HPLOT X(I) + XI,Y(I) + YI
530 I = I + ST: IF I > V THEN I =
     I - V
540  IF P(I) = 1 THEN  GOTO 580
550 P(I) = 1
560  HPLOT  TO X(I) + XI,Y(I) + Y
     I
570  GOTO 530
580  POKE  - 16368,0
590  IF  PEEK ( - 16384) < 128 THEN
     590
600  POKE  - 16368,0
610  REM  LIMPAR AS MATRIZES
620  POKE 109, PEEK (107): POKE 1
     10, PEEK (108)
630  TEXT : POKE 33,39
640  GOTO 290
650  REM
CIRCULO RAPIDO

660 AC = 6
670  HPLOT XI + R - 1,YI
680  FOR I = 0 TO T STEP AC * P: HPLOT
      TO XI + R *  COS (I),YI + R
      *  SIN (I): NEXT : HPLOT  TO
     XI + R - 1,YI
690  RETURN
```

Tente Estas!

Tente rodá-lo no TK-90X e depois modifique-o para conseguir as mais estranhas formas.

```
10 FOR f=68 TO 173 STEP 15
20 READ l: INK l
30 PLOT f,40
40 DRAW 50,87,-PI/3
50 DRAW 4,-7,-PI
60 DRAW -100,0,-PI/3
70 DRAW 4,7,-PI
80 DRAW 50,-87,-PI/3
90 DRAW -8,0,-PI
100 NEXT f
110 DATA 1,2,3,4,4,3,2,1
```

Coleção O Computador na Escola

Servir de suporte didático e de apoio para aplicações dos recursos da informática, nas salas de aula é a principal função e objetivo desta coleção.

Como o título muito bem diz - "O Computador na Escola" é uma coletânea de seis livros que reúne: "Aulas de Matemática"; "Aulas de Matemática Financeira"; "Logo: Teoria e Prática"; "Linguagem BASIC e Programas para Matemática"; "Aulas de Química"; e "Aulas de Física".

Segundo os seus autores (Vicente Paz Fernandez, Antonio Nicolau Youssef, Márcio Pugliesi, Hor Chi Yuen, Fernando José de Almeida, Maria do Carmo Mendonça e Eduardo Castor), a partir da presença quase que diária do computador na escola ou em casa, este "rapidamente vai assumindo um papel importante no processo ensino-aprendizagem, trazendo ao ato de estudar uma nova motivação."

Desta forma, torna-se premente uma publicação do nível e intuito destas obras.

No início de cada livro, os autores dissertam sobre a potencialidade da informática na educação e afirmam que o aluno, ao executar programas, previamente elaborados, ele pode concentrar os seus esforços no desenvolvimento do raciocínio lógico, "deixando para o computador o trabalho de execução de cálculos".

Em suma, o estudante passa de mero ouvinte (espectador passivo) a um participante ativo do aprendizado, pois "passa a não se limitar a questões preparadas, com valores inteiros conhecidos, podendo resolver também problemas com valores e medidas que se aproximam mais, daqueles, da realidade imediata".

Como exemplo do que foi dito acima, está a própria utilização do microcomputador no aprendizado da Matemática, da Física, da Química e de outras matérias. Esta relação com a microcomputação está patente em um dos livros desta coleção ("Linguagem BASIC e Programas para Matemáticas"), onde através do ensinamento do BASIC são introduzidos conceitos de Matemática.

Os leitores, ao adquirirem esta Coleção, não irão precisar seguir uma ordem para lê-los, já que cada livro trata de um assunto diferente. No entanto, para falarmos sobre eles determinamos uma ordenação aleatória.

O primeiro deles é "Aulas de Matemática" que aborda uma série de temas que vão desde conjuntos, funções, trigonometria, progressões à análise combinatória, polinômios e números complexos.

Ao final de cada tópico, os autores apresentam uma seqüência de exercícios que serão resolvidos por meio de programas.

Juros simples, desconto simples, juros compostos, anuidades, sistemas de amortização, fluxo de caixa, inflação e taxas, análise de balanço etc. Estes assuntos poderão ser encontrados no "Aulas de Matemática Financeira", além dos exercícios no final de cada tópico.

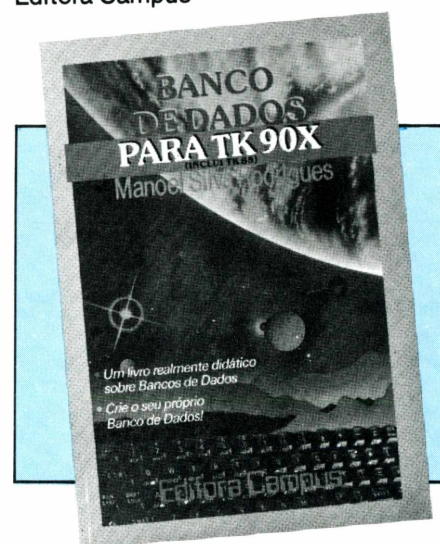
"Logo: Teoria e Prática" é o terceiro livro da série que tem o intuito de fornecer as ferramentas básicas para o uso da linguagem Logo no computador. Nas suas 151 páginas, o usuário encontrará: os comandos iniciais, programação com o Logo, o uso do disco, atividades com o Logo etc.

Dividido em duas partes, o "Linguagem BASIC e Programas para Matemática", primeiramente, oferecem ao leitor noções de: processamento de dados, introdução ao BASIC, comandos, utilização de funções etc. Na parte dois o assunto é Matemática, onde esta ciência é ensinada com o auxílio da informática.

Os dois últimos livros, "Aulas de Química" e "Aulas de Física", seguem o mesmo esquema dos anteriores. O de Química analisa, entre tantos pontos: Ligações, Cinética Química, Titulometria, Termoquímica e Eletroquímica. Já o de Física, o leitor se deparará com os seguintes assuntos: Mecânica, Termologia, Óptica e Eletricidade.

Vale dizer ainda que, no final de cada obra, uma tabela é apresentada com a conversão dos programas para as linhas TRS-80, Sinclair e Apple. ***M.R.*** ◇

Banco de Dados para o TK-90X

Manoel Silva Rodrigues
Editora Campus

Para aproveitar ao máximo este texto, são imprescindíveis: "um microcomputador que use lógica Sinclair, qualquer um com um mínimo de 16K de memória RAM e um gravador. Por parte do leitor, conhecimento elementar de operação do microcomputador em questão e da linguagem BASIC."

É assim que o autor começa a sua obra, dando uma orientação e expondo os pré-requisitos aos seus leitores, inerentes a esta publicação.

"Banco de Dados para o TK-90X" é um livro que ensina a exploração profissional do TK-90X, direcionando o leitor para a criação de um programa em BASIC, que simula um Banco de Dados on-line, monousuário e conversacional.

Extremamente didático e voltado para os iniciantes, este livro (ao seu término) dá ao usuário um total domínio sobre o programa apresentado (a ponto de serem possíveis alterações e complementações do leitor).

Por ser um trabalho essencialmente prático, torna-se necessária a utilização concomitante do microcomputador, para acompanhá-lo.

Esta obra aborda: estrutura do DB, constituição dos arquivos; criação de um menu de opções; eliminação de arquivos; salvar programas; memória ainda disponível; chegada aos arquivos e registros; recuperação de dados; cálculo de salário líquido; inclusão de registros em arquivos já criados; ordenação de arquivo; imple-

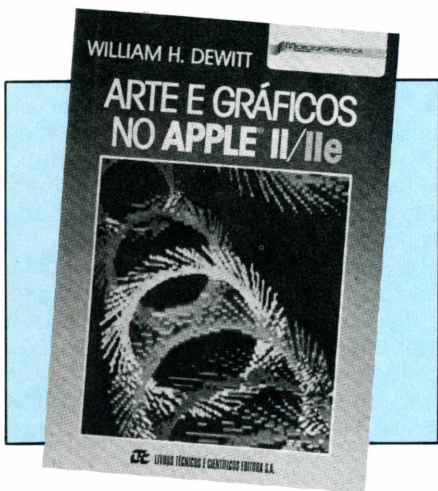
LIVROS

mentação de diretório; salvamento e recuperação do DB e de seu diretório.

Segundo o seu autor, com este livro, o leitor poderá ver o programa crescer aos poucos, acompanhando o seu desenvolvimento e desta forma, servirá também como um modelo de idealização de outros programas.

Manoel Silva Rodrigues é professor de Microinformática da Faculdade Simonsen, Analista de Sistemas da Rede Ferroviária Federal e colaborador de diversas revistas especializadas. *M.R.*

Arte e Gráficos no Apple II/IIe

William H. Dewitt
Editora Livros Técnicos e Científicos

"Lembre-se, o computador é o índio, você é o chefe!"

É com esta afirmação que o autor encoraja os seus leitores a entrarem na área de computação gráfica.

Dirigido aos usuários de computadores da linha Apple, esta publicação é uma introdução para iniciantes em gráficos. No entanto, William Dewitt parte do pressuposto que o leitor já tenha experiência em programação BASIC, seja familiarizado com o "Applesoft Tutorial" e "Applesoft Basic Programming Training Manual"; que esteja operando com 48K de memória; use um ou mais disk-drives e possua um monitor colorido.

Com a finalidade de ser um manual para aqueles que queiram criar imagens nos seus Apples, "Artes e Gráficos no Apple II/IIe" atinge o seu objetivo.

No primeiro capítulo são apresentadas as "ferramentas" necessárias (operacionais e de programação) para o usuário escrever programas gráficos corretamente. Isto envolve números randômicos e esquemas de cálculos, além da seqüência e outras convenções.

Comandos gráficos de baixa resolução e 15 programas ilustrativos, que demonstram o "uso de comandos de baixa resolução na geração de imagens que vão desde um simples ponto até um fluxograma dinâmico", são expostos nos capítulos dois e três.

Já os capítulos quatro e cinco mostram os comandos gráficos de alta resolução.

A criação de formas e tabelas sem a utilização de números binários ou hexadecimais é exibida nos capítulos seis, sete e oito.

No capítulo nove são postas em prática as técnicas de programação, anteriormente aprendidas, para a criação de imagens no computador.

Os leitores podem aprender ainda, com este livro, como armazenar "figuras completas" para recuperação e exibição como um "slide show" e como obter cópias das imagens desenvolvidas.

Num total de 60 programas, que ilustram todas as técnicas apresentadas no decorrer do livro, além de fotos e desenhos, o leitor com isto encontrará um bom subsídio para soltar a sua imaginação e criatividade para conquistar o universo dos gráficos de alta e baixa resolução.

William Dewitt é presidente da Computer Art., lecionou gráficos na Universidade de Rochester e é membro do Institute of Fellows, Rochester Institute of Technology. *M.R.*

Outros Lançamentos

MUMPS - Guia do Usuário

Editora McGraw-Hill

dBASE III - Guia do Usuário

Edward Jones
Editora McGraw-Hill

TRS-80 - Programação Usando Arquivos de Dados

Le Roy Finkel - Jerald Brown
Livros Técnicos e Editora - LTC.

Livros mais vendidos no mês de Julho/86

Profissionais:

- Aprofundando-se no MSX
  Editora Aleph
- Lotus 1-2-3 - Guia do Usuário
  Editora McGraw-Hill
- Redes locais de computadores - Tecnologia e Aplicações
  Editora McGraw-Hill
- MS-DOS - Guia do Usuário
  Editora McGraw-Hill
- Basic Avançado para o TK-90X
  Editora McGraw-Hill

Universitários:

- IBM-PC Programação Basic - Guia do Usuário - Editora McGraw-Hill
- dBase II Soluções para microcomputadores - Editora Atlas
- Introdução ao Sistema Banco de Dados - Editora Campus

Livrarias consultadas:

São Paulo: ***Antena, Saraiva, Siciliano, Poliedro, Litec, Manduri, Teixeira, 5.ª Avenida.***
Rio de Janeiro: ***Interciência***

Às livrarias de outros estados solicitamos que nos enviem sua pesquisa.

Aula 5

Daniel José Burd

Nesta lição aprenderemos a enviar dados para o gravador e a recebê-los. Veremos também, as rotinas da ROM que geram sons.

Manipulando o gravador cassete

Quando mandamos o computador enviar dados para o gravador, através do comando SAVE, dois blocos de bytes são deslocados. O primeiro bloco, conhecido como ***header,*** contém as seguintes informações:

- tipo de programa (exemplo: BASIC);
- nome do programa;
- tamanho;
- endereço inicial.

O segundo bloco é o programa em sí.

Dessa forma, ao executarmos um LOAD, o primeiro bloco a ser lido será o ***header,*** o qual informará ao micro onde o programa deve ser armazenado, o nome do programa etc.

A rotina LOAD DATA

Quando desejamos ler dados do gravador através de instruções Assembler, devemos dar várias informações ao computador. O computador necessita saber se o que ele vai receber é um ***header*** ou não, onde colocar o bloco de bytes, qual o tamanho do bloco etc. Para informamos todos esses dados ao micro devemos usar a seguinte convenção:

acumulador - se for 0, implica em leitura ou escrita de ***header;***
se for 255 implica em leitura ou escrita de bloco:

flag de carry - se for 0 implica em verificação de dados (VERIFY);
se for 1 implica na leitura de dados (LOAD);

registradores IX - contêm o endereço a partir do qual devem ser colocados (ou enviados) os bytes;

registradores DE - contêm o tamanho do bloco.

Após especificarmos todos os parâmetros, basta dar um CALL LOAD DATA (0556h) e pronto. Aqui estão as possíveis combinações dos parâmetros anteriores:

```
LOAD BYTES: 1d de,nbytes      ;nbytes = tamanho do bloco
            1d ix,início
            1d a,255          ;não header
            scf               ;LOAD
            call LOAD DATA
            ret

LOAD HEADER: 1d de,17         ;17 bytes e o tamanho do
                              header
             1d ix, ínico do header
             xor a
             scf
             call LOAD DATA
             ret

VERIFYBYTES: 1d de,nbytes
             1d ix,ínicio
             1d a,255
             and a
             call LOAD DATA
             ret

VERIFYHEADER: 1d de,17
              1d ix,ínicio do header
              xor a
              call LOAD DATA
              ret
```

A rotina SAVE DATA (04C2h)

Essa rotina usa os mesmos parâmetros da LOAD DATA exceto a flag de carry. Aqui estão os possíveis usos dessa rotina:

```
SAVEBYTES: 1d de,nbytes
           1d ix, ínicio    ;ínicio do bloco a ser salvo
           1d a,255
           call SAVE DATA
           ret

SAVEHEADER: 1d de,17
            1d ix,ínicio do header
            xor a            ;a = 0 implica em header
            call SAVE DATA
            ret
```

Vamos, a título de exemplo, salvar os bytes que compõem a imagem desenhada na tela. Como sabemos, de lições anteriores, a região ocupada pela tela inicia-se no endereço 16384 e tern comprimento 6912 bytes, incluindo a região de atributos.

Dessa forma, um programa que salva essa região é mostrado abaixo:

```
30000 11 00 18      1d de,6912      ;salva 6912 bytes;
30003 DD 21 00 40   1d ix,16384     ;aponta para o arquivo da
                                     tela;
30007 3E FF         1d a,255        ;indica que não é header;
30009 CD C2 04      call SAVE DATA
30011 C9            ret
```

Simples não! Para usá-lo, "suje" a tela com qualquer coisa (exemplo: PRINT AT 10,10;"TELA 1"). Digite RAND USR 30000, não "dê" ENTER ainda, primeiro ligue o gravador (RECORD), e só depois dê ENTER.

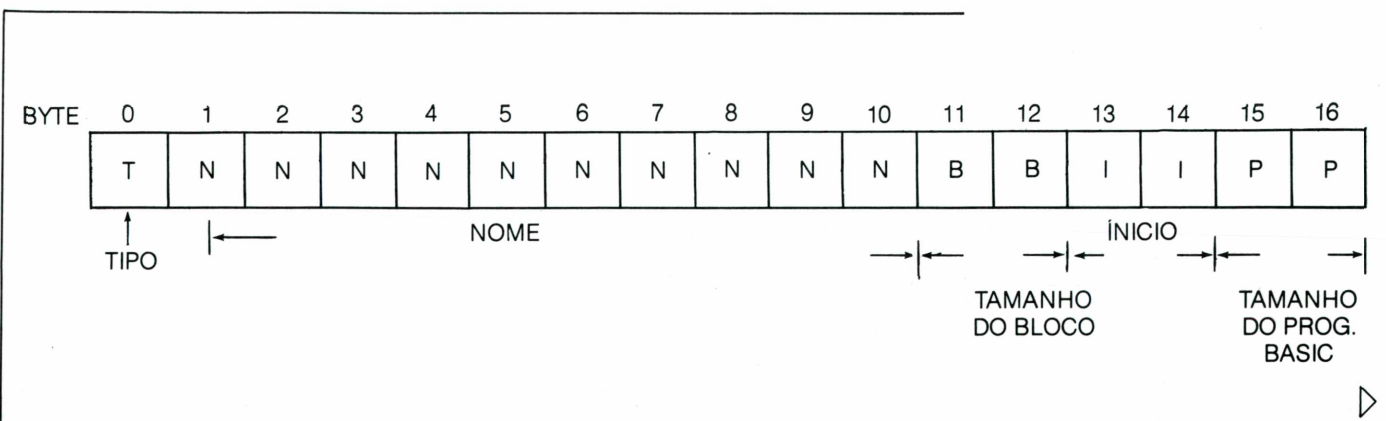
Vamos agora "LOADar" a nossa tela:

```
30500 11 00 18      1d de,6912
30503 DD 21 00 40   1d ix,16384
30507 3E FF         1d a,255
30509 37            scf
30510 CD 56 05      call LOAD DATA
30513 C9            ret
```

Agora ligue o gravador (PLAY), logicamente, após rebobinar a fita) e só então "dê" RAND USR 30500. Aí está a sua tela!

Você deve estar curioso quanto aos 17 bytes do ***header.*** Veremos agora como criá-lo e lê-lo:

O HEADER

O ***header*** possui o seguinte formato:

A ROM DO TK-90X

O byte 0 indica o tipo de programa:
0 - implica em programa BASIC;
1 - matriz numérica;
2 - matriz alfanumérica;
3 - bloco de dados.
os bytes de 1 a 10 contêm o nome do programa;
os bytes 11 e 12 contêm o tamanho do bloco se o byte 0 for 3;
os bytes 13 e 14 contêm o endereço inicial do bloco ou linha inicial do programa BASIC.
os bytes 15 e 16 contêm o tamanho da área ocupada pelo programa BASIC.
O programa referente às listagens A, B e C lê e exibe os dados do ***header*** de qualquer programa. Procure entendê-lo. Vale a pena!

```
30000 16 00 05 4C 45 49 54 4F
30008 52 20 44 45 20 48 45 41
30016 44 45 52 16 01 05 70 6F
30024 72 20 44 61 6E 69 65 6C
30032 20 4A 6F 73 65 20 42 75
30040 72 64 2E 16 0A 08 12 01
30048 45 53 50 45 52 41 4E 44
30056 4F 20 48 45 41 44 45 52
30064 2E 16 05 00 50 52 4F 47
30072 52 41 4D 41 3A 16 07 00
30080 54 49 50 4F 3A 16 09 00
30088 54 41 4D 41 4E 48 4F 3A
30096 16 0B 00 49 4E 49 43 49
30104 4F 2F 4C 49 4E 48 41 3A
30112 16 0D 00 54 41 4D 41 4E
30120 48 4F 20 42 41 53 49 43
30128 3A 30 35 37 36 38 62 61
30136 73 69 63 6D 2E 6E 75 6D
30144 6D 2E 61 6C 66 62 6C 6F
30152 63 6F 10 27 E8 03 64 00
30160 0A 00 01 00 00 41 53 53
30168 45 4D 42 4C 45 52 20 88
30176 16 00 00 88 16 00 00 00
30184 00 00 00 00 00 00 21 00
30192 40 11 01 40 01 FF 17 36
```

```
30200 00 ED B0 23 13 01 FF 02
30208 36 20 ED B0 C9 01 B1 75
30216 DD 21 CA 75 DD 56 01 DD
30224 5E 00 3E 2F A7 3C ED 52
30232 30 FB 19 02 DD 23 DD 23
30240 03 1D 20 E8 11 B1 75 01
30248 05 00 CD 3C 20 C9 11 11
30256 00 DD 21 D4 75 AF 37 CD
30264 56 05 C9 3E 02 CD 01 16
30272 CD EE 75 11 30 75 01 41
30280 00 CD 3C 20 CD 2E 76 CD
30288 EE 75 3E 12 D7 3E 00 D7
30296 11 71 75 01 0C 00 CD 3C
30304 20 11 D5 75 01 0A 00 CD
30312 3C 20 11 7D 75 01 08 00
30320 CD 3C 20 3A D4 75 5F CB
30328 27 CB 27 83 21 B6 75 5F
30336 16 00 19 EB 01 05 00 CD
30344 3C 20 11 85 75 01 0B 00
30352 CD 3C 20 2A DF 75 CD 05
30360 76 01 10 00 11 90 75 CD
30368 3C 20 2A E1 75 CD 05 76
30376 01 11 00 11 A0 75 CD 3C
30384 20 2A E3 75 CD 05 76 C9
00
```

LISTAGEM B

```
30000  MESS 1:  DEFM  22,0,5, "LEITOR DE HEADER"
30019           DEFM  22,1,5, "POR DANIEL JOSÉ BURD."
300H3           DEFM  22,10,8,18,1, "ESPERANDO HEADER."
30065  MESS 2:  DEFM  22,5,0, "PROGRAMA:"
30077  MESS 3:  DEFM  22,7,0, "TIPO:"
30085  MESS 5:  DEFM  22,9,0, "TAMANHO:"
30096  MESS 6:  DEFM  22,11,0, "INÍCIO/LINHA:"
30112  MESS 7:  DEFM  22,13,0, "TAMANHO BASIC:"
30129  BUF:     DEFb  0,0,0,0,0, ←—— BUFFER P/ IMPRESSÃO.
30134  MESS 8:  DEFM  "BASIC"; "M.NUM", "M,ALF", "BLOCO"
30154  BASE:    DEFW  10000,1000,100,10,1 ←— BASE DECIMAL
30164  HEADER:  DEFb  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, ←—┐
                      └—→ armazena o header
```

LISTAGEM C

```
30190 - CLS LD HL - 16384  ┐
            LD DE - 16385  │
            LD BC - 6143   ├ LIMPA A TELA
            LD (HL),0      │
            LDIR           ┘
            INC HL´        ┐
            INC DE         │
            LD BC - 767    ├ COLORE A TELA DE VERDE
            LD (HL),32     │
            LDIR           │
            RET            ┘
```

▷

```
30213 - TI:  LD BC - 30129                     ; BUF
             LD IX - 30154                     ; BASE
       F2:   LD D, (IX + 1)
             LD E, (IX + 0)
             LD A, 47
             AND A
       F1:   INC A
             SBC HL,DE
             JR NC, F1
             ADD HL,DE
             LD (BC),A
             INC IX
             INC IX
             INC BC
             DEC E
             JR NZ,F2
             LD DE, 30129
             LD BC,5
             CALL 8252 ; PRINT STRING
             RET

  Essa rotina transforma o conteúdo
  dos registradres HL em string e depois
  o imprime.

30254 - LEHEAD: LD DE, 17
                LD IX, 30164          ; HEADER
                XOR A
                SCF
                CALL 1366             ; LOAD DATA
                RET
                                      } LE HEADER

30267 - INÍCIO: LD A,2
                CALL 5633             ; ABRE O CANAL 5
                CALL CLS              ; 30190
                LD DE, MESS 1
                LD BC, 65             ; 65 = COMPRIMENTO DA MESS 1   } IMPRIME A MESS 1
                CALL PRINT STRING, 8252
                CALL LEHEAD           ; LEO HEADER
                CALL CLS
                LD A,18
                RST 16                } FLASH 0
                LDA,0
                RST 16

                LD DE, MESS 2
                LD BC, 12                             } IMPRIME A MESS 2
                CALL 8252         ; PRINT STRING
                LD DE, 30165      ; INÍCIO DO NOME DO PRG
                LD BC, 10                             } IMPRIME O NOME DO PROG.
                CALL 8252
                LD DE, MESS 3
                LD BC, 8          } IMPRIME A MESS 3
                CALL 8252
                LDA, (30164)      ; A CONTÉM O TIPO DE PROGRAMA
                LD e,A
                SLA A             } MULTIPLICA A*5
                SLA A
                ADD A,e
                LD HL, MESS 8
                LD e,A            } HL APONTA P/.O TIPO DO PROG.
                LD D,0
                ADD HL,DE
                EX DE, HL
                LD BC,5           } IMPRIME O TIPO DE PROG.
                CALL 8252
                LD DE, MESS 5
                LD BC, 11         } IMPRIME A MESS 5
                CALL 8252
                LD HL, (10175)    ; HL CONTÉM O TAMANHO DO PROG.  } IMPRIME O
                CALL TI                                             TAMANHO
```

▷

A ROM DO TK-90X

```
LD BC, 16       }
LD DE, MESS 6   }  IMPRIME A MESS 6
CALL 8252       }
LD HL, (30177)     ; HL - INÍCIO OU LINHA INICIAL  }  IMPRIME O
CALL TI                                            }  INÍCIO PROG.
LD BC, 17       }
LD DE, MESS 7   }  IMPRIME A MESS 7
CALL 8252       }
LD HL, (30179)     ; HL CONTÉM O TAMANHO DO PROG. BASIC  }  IMPRIME
CALL TI                                                  }  TAMANHO
RET                                                      }  BASIC
```

Tabela B

Nota		*Freqüência (Hz)*
DÓ	C 0	16.35
	C # 0	17.32
RÉ	D 0	18.35
	D # 0	19.44
MI	E 0	20.60
FÁ	F 0	21.83
	F # 0	23.12
SOL	G 0	24.50
	G # 0	25.96
LÁ	A 0	27.50
	A # 0	29.14
SI	B 0	30.87
	C 1	32.70
	C # 1	34.64
	D 1	36.70
	D # 1	38.88
	E 1	41.20
	F 1	43.66
	F # 1	46.24
	G 1	49.00
	G # 1	51.92

Som

Existem duas formas principais de executarmos sons com o auxílio da ROM do TK-90X.

A primeira usa a rotina BEEPER, e a segunda a rotina BEEP (ou SOUND) de modo muito semelhante ao BASIC. Veremos como usar cada uma:

A rotina BEEPER (0385h) necessita, como dados de entrada, a freqüência da nota e a que duração. A freqüência fica nos registradores HL e a duração nos DE, por exemplo:

O comando SOUND 1,0 pode ser simulado por:

```
30000 11 05 01    1d de,duração
30003 21 6A 06    1d h1,nota
30006 CD B5 03    call BEEPER
30009 C9          ret
```

Para calcularmos a duração e a nota, usam-se os seguintes algorítmos:
duração = freqüência * tempo
nota = 437500/freqüência -30.125

Verifique que a freqüência de um DO ou C4 é 261.60 Hz (dada na tabela B). Se substituirmos esse valor nas equações acima, para tempo igual a um segundo, teremos os valores de HL e DE. Verifique isso através do programa abaixo:

```
10 CLS
20 INPUT"FREQUENCIA (Hz) = ";F
30 INPUT"DURAÇÃO(s) = ";D
40 CLS
50 PRINT"HL = ";437500/F-30.125
60 PRINT"DE = ";F*D
70 PAUSE 9999
80 GOTO 10
```

Bom divertimento a todos.

Tente Estas!
TK 90X

```
 1 INPUT a$: LET c=LEN
(a$)-1: PRINT a$
  8 LET a=16384
  9 FOR m=0 TO 7
 10 FOR i=a TO a+c
 20 POKE i,0
 30 NEXT i
 40 LET a=a+256
 50 NEXT m
```

```
 10 INPUT "Linha:";l;"
Coluna:";c;" Raio:";r
 20 FOR i=1 TO 102
 30 LET a=c+r*SIN (i/8)
 LET b= l+r*COS (i/8)
 40 PLOT a,b
 50 NEXT i
```

A Microdigital lança no Brasil o micro pessoal de maior sucesso no mundo.

A partir de agora a história dos micros pessoais vai ser contada em duas partes: antes e depois do TK 90X.

O TK 90X é, simplesmente, o único micro pessoal lançado no Brasil que merece a classificação de "software machine": um caso raro de micro que pela sua facilidade de uso, grandes recursos e preço acessível recebeu a atenção dos criadores de programas e periféricos em todo o mundo.

Para você ter uma idéia, existem mais de 2 mil programas, 70 livros, 30 periféricos e inúmeras revistas de usuários disponíveis para ele internacionalmente.

E aqui o TK 90X já sai com mais de 100 programas, enquanto outros estão em fase final de desenvolvimento para lhe dar mais opções para trabalhar, aprender ou se divertir que com qualquer outro micro.

O TK 90X tem duas versões de memória (de 16 ou 48 K), imagem de alta resolução gráfica com 8 cores, carregamento rápido de programas (controlável pelo próprio monitor), som pela TV, letras maiúsculas e minúsculas e ainda uma exclusividade: acentuação em português.

Faça o seu programa: peça já uma demonstração do novo TK 90X.

MICRODIGITAL

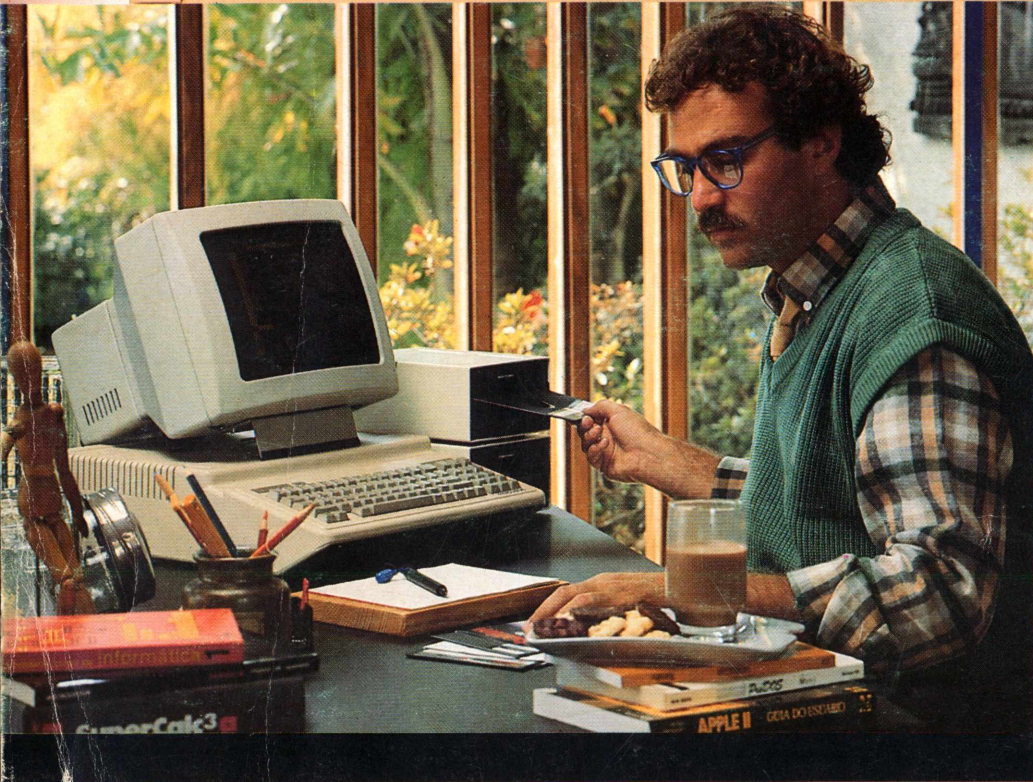
Chegou o micro cheio de programas.*

TK 90X

*Sujeito a alteração sem prévio aviso.

Filiada à ABICOMP

A Microdigital lança o TK 3000 IIe e tira os 8 anos de atraso do micro brasileiro.

FOX

Chega ao Brasil o sucessor do Apple IIe "Enhanced".*

Tire da cabeça tudo que você já teve, tem ou viu em matéria de micros em geral e Apples em particular.*
É a primeira vez que chega ao Brasil um Apple último modelo: o TK 3000 IIe é uma versão ainda mais avançada do avançadíssimo Apple IIe* "Enhanced", lançado em maio de 1985 nos Estados Unidos.*
É o único que roda Totalworks e Supercalc 3a, entre milhares de outros. Faz em segundos o que os demais micros levam intermináveis minutos para fazer, tem memória básica de 64 Kbytes (expandível com placas até 1 Megabyte) e teclado numérico incorporado.
Escreve em português com todas as letras e acentos e com maior facilidade do que uma máquina de escrever eletrônica.
E, entre outras exclusividades, tem um design anatômico, para maior conforto do operador e produtividade no trabalho.
Venha logo conhecer e reservar o seu TK 3000 IIe nos Revendedores Autorizados Microdigital.
Os 8 anos-tecnologia que o separam dos outros micros, podem ser exatamente a distância que você vai colocar entre sua empresa e os concorrentes.